

Deep learning for anisoplanatic optical turbulence mitigation in long-range imaging

Matthew A. Hoffmire^a,^{*} Russell C. Hardie^{a,*}, Michael A. Rucci,^b
Richard Van Hook,^b and Barry K. Karch^b

^aUniversity of Dayton, Department of Electrical and Computer Engineering, Dayton, Ohio,
United States

^bAir Force Research Laboratory, AFRL/Rymt, Wright-Patterson AFB, Ohio,
United States

Abstract. We present a deep learning approach for restoring images degraded by atmospheric optical turbulence. We consider the case of terrestrial imaging over long ranges with a wide field-of-view. This produces an anisoplanatic imaging scenario where turbulence warping and blurring vary spatially across the image. The proposed turbulence mitigation (TM) method assumes that a sequence of short-exposure images is acquired. A block matching (BM) registration algorithm is applied to the observed frames for dewarping, and the resulting images are averaged. A convolutional neural network (CNN) is then employed to perform spatially adaptive restoration. We refer to the proposed TM algorithm as the block matching and CNN (BM-CNN) method. Training the CNN is accomplished using simulated data from a fast turbulence simulation tool capable of producing a large amount of degraded imagery from declared truth images rapidly. Testing is done using independent data simulated with a different well-validated numerical wave-propagation simulator. Our proposed BM-CNN TM method is evaluated in a number of experiments using quantitative metrics. The quantitative analysis is made possible by virtue of having truth imagery from the simulations. A number of restored images are provided for subjective evaluation. We demonstrate that the BM-CNN TM method outperforms the benchmark methods in the scenarios tested. © The Authors. Published by SPIE under a Creative Commons Attribution 4.0 Unported License. Distribution or reproduction of this work in whole or in part requires full attribution of the original publication, including its DOI. [DOI: [10.1117/1.OE.60.3.033103](https://doi.org/10.1117/1.OE.60.3.033103)]

Keywords: atmospheric turbulence; turbulence mitigation; deep learning; convolutional neural network.

Paper 20210090 received Jan. 23, 2021; accepted for publication Mar. 3, 2021; published online Mar. 23, 2021.

1 Introduction

The acquisition of long-range images is often affected by atmospheric optical turbulence. The turbulence degradation is caused by changes in temperature and pressure along the optical path that leads to variations in the index of refraction. The result is a quasi-periodic spatial and temporal blur and warping.¹⁻⁴ In the field of astronomy, this has been heavily investigated.³ For astronomy, the field-of-view tends to be narrow, which causes the scene to be affected by the atmosphere evenly across the entirety of the image. In this isoplanatic scenario, one can model the degradation with a single linear shift-invariant (LSI) point spread function (PSF). In this paper, we consider a terrestrial long-range imaging scenario with a relatively wide field-of-view. This leads to anisoplanatic conditions where the degradation varies across the scene. This requires a spatially varying PSF to successfully model the degradation.

There are many different approaches proposed in the literature to address the problem of turbulence mitigation (TM) via post processing of degraded images. Most methods are based on processing a sequence of short-exposure (SE) images.^{1,5} The reasoning is that SE images can effectively freeze warping distortions temporally. This produces images with minimal motion blur. However, the geometry of these individual frames varies temporally and is most likely

*Address all correspondence to Russell C. Hardie, rhardie1@udayton.edu

incorrect.^{6,7} On the other hand, averaging the SE frames produces an image that is equivalent to a long-exposure (LE) image. The LE image tends to have the correct geometry, but also has significant amounts of motion blur.¹ For these reasons, many TM algorithms follow the steps of dewarping, fusion, and then blind deconvolution.^{1,8–15} Dewarping refers to motion compensation of the SE frames so as to reduce the effects of motion blur when fused. The motion compensation is typically followed by some sort of averaging to combine the frames and reduce noise. This fusion also serves to average the spatially varying blur so that the resulting blur can be well modeled by a simpler LSI PSF.¹ Finally, the deconvolution step attempts to undo the blurring from the LSI PSF.

The block matching and Wiener Filtering (BM-WF)¹ method uses a parametric PSF model and a fixed Wiener filter (WF) for the deconvolution. The parametric model for the PSF takes into account diffraction from the optics, SE blurring, and residual motion blur from imperfect image registration using BM. The parametric PSF used in BM-WF has only two unknown parameters, making it much easier to optimize than traditional blind deconvolution methods. Another popular TM method is bispectral speckle imaging (BSI).^{8–10,16–23} The BSI method attempts to mitigate turbulence through the estimation of image magnitude and phase separately. The name comes from “high-frequency intensity fluctuations that are referred to as speckles.”⁹ The BM-WF performed well in the studies reported by Hardie et al.¹ and it is well suited to adapting to machine learning. For these reasons, we have chosen to build on the framework of BM-WF here.

Designing, optimizing, and evaluating TM algorithms are complicated by the difficulty in obtaining ground truth imagery. Recently, there have been a number of turbulence simulators for anisoplanatic imaging presented in the literature.^{2,19,24–27} Simulation has the advantage of being able to produce a virtually unlimited amount of degraded images with corresponding ground truth for nearly any scenario. These simulated data can be used for training machine learning methods and for quantitative performance analysis of any TM methods. We believe accurate simulation tools are essential to advance the field of TM. One well-validated turbulence simulator is the numerical wave-propagation anisoplanatic simulator developed by Hardie et al.² This is the method we shall use for testing the TM algorithms presented in this paper. We consider this type of numerical wave method the “gold standard” for anisoplanatic turbulence simulation. However, it does have a relatively high computational complexity. Thus, we also consider a fast simulation method based on that in Schwartzman et al.²⁷ We have extended Schwartzman’s warping-only simulator to add atmospheric blurring, and we use different tilt correlation statistics as described in Sec. 2. Machine learning TM methods require a large amount of training data. Thus, we believe our fast simulator is an effective choice for this purpose. While we use the fast simulator for training, we shall validate all of our results by generating testing data using the numerical wave-propagation simulator.

The WF used for deconvolution in the BM-WF method is known to be optimum in a mean squared error sense for wide sense stationary (WSS) Gaussian images and noise along with an LSI blur. However, in practice, we know that real images are not generally WSS and the statistics are not necessarily Gaussian. Furthermore, while the fusion process does seek to make the residual blur LSI, it may not be perfectly successful in this regard. For these reasons, we believe that there is room for improvement in the BM-WF approach with respect to the WF. It is our hypothesis that we can replace the WF with a convolutional neural network (CNN) and use deep learning (DL) to obtain improved performance relative to the BM-WF method. We argue that the non-linear CNN method is better equipped to address the non-stationary nature of most images and also to address any spatially varying residual blur that may be present. The availability of the new atmospheric turbulence simulators described above makes it possible to generate a virtually unlimited amount of training data to support successful DL with a CNN in this application. We refer to the new proposed algorithm as the BM and convolutional neural network (BM-CNN) method.

Recently CNNs have been used heavily in image processing for a variety of tasks including image restoration and image super resolution (SR). There have been many studies done that show CNNs are successful in restoring images.^{28–32} While there are many different types of networks that can do SR and restoration,^{33–37} here, we employ one recently proposed by Elwarfalli and Hardie.³⁸ The network is referred to as the Fusion of Interpolated Frames

Network (FIFNet). This network has been chosen based on good demonstrated SR performance and its lightweight architecture, i.e., it has relatively few convolution layers and can be trained from scratch rather quickly. Large networks can take days or even weeks to train from scratch. FIFNet can be trained in a matter of hours. The larger networks also often require transfer learning where pretrained networks are imported and refined with further training. Such transfer learning is not necessary for FIFNet.

To the best of our knowledge, there is only one prior study in the literature that has been done using CNNs to perform TM.³⁹ In that work, they show that a CNN can restore images affected by turbulence. However, their experimental study uses a simple simulator with images degraded by only nine different PSFs. Furthermore, they do not employ any image registration in their restoration method. In contrast, here, we use a full anisoplanatic simulator² for validation, and a modified version of the Schwartzman's warping simulator for training.²⁷ Our BM-CNN method also gains improved performance by incorporating BM registration prior to CNN processing. We study the performance of our BM-CNN method on four different turbulence levels at two noise levels. We shall show there that the BM-CNN consistently outperforms the BM-WF method and other benchmark methods in the scenarios tested here. We find the boost in performance most pronounced in light to moderate turbulence and moderate to high noise levels.

The rest of the paper is organized as follows. Section 2 provides a description of turbulence characterization statistics. It also explains the two turbulence simulation methods employed here. Section 3 introduces the TM methods studied in this paper. Specifically, we introduce our proposed BM-CNN method and compare and contrast it with the BM-WF benchmark method. Details of the CNN architecture and training processes are also provided in Sec. 3. Section 4 presents the experimental results that includes a detailed quantitative analysis. Finally, conclusions are articulated in Sec. 5.

2 Turbulence Simulation

2.1 Turbulence Characterization

Atmospheric optical turbulence impacts the imaging process by introducing random pockets of air with varying indices of refraction along the optical path. Variations in the index of refraction are often modeled with a refractive index structure function. This is given as

$$D_n(\mathbf{r}) = \langle [n(\mathbf{x} + \mathbf{r}) - n(\mathbf{x})]^2 \rangle = C_n^2 r^{2/3}, \quad (1)$$

where \mathbf{x} and \mathbf{r} are three-dimensional spatial coordinate vectors, $r = |\mathbf{r}|$, $n(\cdot)$ is the index of refraction, and $\langle \cdot \rangle$ represents an ensemble mean operator. The parameter C_n^2 is the refractive index structure parameter and it has units of $\text{m}^{-2/3}$. Typical values of C_n^2 tend to range from 1×10^{-13} to $1 \times 10^{-17} \text{ m}^{-2/3}$.⁴⁰ When this quantity varies along the optical path, it may be expressed as a function, $C_n^2(z)$, and here we define z as the distance from the source.

Another key turbulence statistic that takes the full optical path into account is called the Fried parameter and is denoted r_0 . The Fried parameter is also known as the **atmospheric coherence diameter**.^{3,40} This parameter can be expressed in terms of the refractive index structure function as

$$r_0 = \left[0.423 \left(\frac{2\pi}{\lambda} \right)^2 \int_{z=0}^{z=L} C_n^2(z) \left(\frac{z}{L} \right)^{5/3} dz \right]^{-3/5}, \quad (2)$$

where λ is the wavelength and the path length is given by the variable L . The Fried parameter is the key parameter that governs the atmospheric optical transfer function (OTF), the PSF, and point source tilt variance.⁴¹ It is worth noting that **a small r_0 relative to the camera aperture indicates a high level of atmospheric turbulence**.

Warping from turbulence can be characterized by the point source angular tilt variance. The two-axis Zernike tilt (Z-tilt) variance in units of radians squared is given by Tyson⁴² in terms of r_0 as

Table 1 Optical parameters used in simulation results.

Parameter	Value
Aperture	$D = 0.2034$ (m)
Focal length	$l = 1.2000$ (m)
F-number	5.8997
Wavelength	$\lambda = 0.5250$ (μm)
Object distance	$L = 7000$ (m)
Nyquist pixel spacing	$\delta = 1.5487$ (μm)
Cutoff frequency	$\rho_c = 322.8571$ (cycles/mm)

$$T_Z^2 = .3641 \left(\frac{D}{r_0} \right)^{5/3} \left(\frac{\lambda}{D} \right)^2, \quad (3)$$

where D is the aperture of the optics. As turbulence strength increases, we see a smaller r_0 and increased tilt variance. The increase in tilt variance corresponds into increased warping in the observed imagery. There is also an increase in blurring as governed by the OTF.

Another relevant statistic is the isoplanatic angle.³ Point sources separated by less than this angle will have similar PSFs and a mean wave function phase difference at the aperture of <1 rad.^{40,43} The isoplanatic angle can also be expressed as a weighted integral of $C_n^2(z)$, yielding^{40,43}

$$\theta_0 = \left[2.91 k^2 L^{5/3} \int_{z=0}^{z=L} C_n^2(z) \left(1 - \frac{z}{L} \right)^{5/3} dz \right]^{-3/5}. \quad (4)$$

Generally speaking, as the turbulence level increases, the isoplanatic patch size decreases. This means the warping is less correlated spatially over a given separation distance. This tends to make it harder to perform image registration.

The optical system parameters used for the analysis and results in this paper are listed in Table 1. The specific turbulence statistics are listed in Table 2. All of the experimental results presented in Sec. 4 are based on these values. We consider a visible system operating at relatively long range (7 km) at four different turbulence levels. We further assume that the sensor is Nyquist sampled relative to the diffraction-limited optical cutoff frequency. It is worth noting that some prior work has been done addressing the problems of turbulence and undersampling jointly using an approach based on the BM-WF method.⁴¹ We forgo that extra complication in this study, as the emphasis here is on using machine learning for TM. The optical and turbulence parameters have been selected to closely match those in the paper² that describes the anisoplanatic simulator employed here. The rationale for this choice is that the simulator is thoroughly validated using

Table 2 Turbulence parameters used in the simulation results.

Parameter	Turbulence degradation			
	Level 1	Level 2	Level 3	Level 4
$C_n^2 \times 10^{-15}$ ($\text{m}^{-2/3}$)	0.0307	0.1227	0.4909	1.9638
Theoretical r_0 (m)	0.3863	0.1681	0.0732	0.0319
Theoretical D/r_0 (unitless)	0.5265	1.2097	2.7789	6.3843
Isoplanatic angle (pixels)	13.4454	5.8516	2.5472	1.1087
RMS tilt (pixels)	0.5000	1	2	4

those parameters.² Thus, we have a high confidence that the data from the anisoplanatic simulator are realistic and consistent with key theoretical metrics.

The OTF for an imaging system in turbulence is well modeled as

$$H(\rho) = H_{atm}(\rho)H_{dif}(\rho), \quad (5)$$

where $\rho = \sqrt{u^2 + v^2}$ and u and v are the spatial frequencies in units of cycles per unit distance. The average atmospheric OTF component is commonly expressed as³

$$H_{atm}(\rho) = \exp\left\{-3.44\left(\frac{\lambda\rho}{r_0}\right)^{5/3}\left[1 - \alpha\left(\frac{\lambda\rho}{D}\right)^{1/3}\right]\right\}, \quad (6)$$

where l is the camera focal length and D is the aperture diameter. The parameter α here serves as a switch where the LE atmospheric OTF is generated by Eq. (6) when $\alpha = 0$. The ensemble-average SE OTF is given by Eq. (6) when $\alpha = 1$. The SE OTF is effectively a shift corrected OTF. When partial tilt correction is achieved by image registration, the parameter $\alpha \in (0,1)$ can be used as a tilt correction factor¹ to produce an OTF that accounts for residual tilt variance motion blurring.

For an imaging system with a circular aperture the diffraction-limited OTF is given as⁴⁴

$$H_{dif}(\rho) = \begin{cases} \frac{2}{\pi} \left[\cos^{-1}\left(\frac{\rho}{\rho_c}\right) - \frac{\rho}{\rho_c} \sqrt{1 - \left(\frac{\rho}{\rho_c}\right)^2} \right] & \rho \leq \rho_c \\ 0 & \text{otherwise} \end{cases}. \quad (7)$$

Here, the variable $\rho_c = 1/(\lambda f_n)$ is the optical cutoff frequency with $f_n = l/D$ being the f -number of the optics. It is worth noting that the theoretical LE PSF is the inverse Fourier transform of Eq. (5) with $\alpha = 0$. Similarly, the theoretical average SE PSF is the inverse Fourier transform of Eq. (5) with $\alpha = 1$.

Examples of OTFs for the optical parameters and turbulence levels in Tables 1 and 2 are shown in Figs. 1 and 2. The LE OTFs are in Fig. 1 and the ensemble-average SE OTFs are in Fig. 2. Note that as C_n^2 increases, the OTF becomes more low pass in nature and has more blurring. Also, the LE OTFs have more blurring than the corresponding SE OTFs for the same turbulence level. It is worth noting that additional OTF components could be added to the model. These components could include defocus, optical aberrations, and atmospheric aerosol contributions. In future work, the addition of aerosol OTFs^{45–47} may be a particularly interesting study for terrestrial applications.

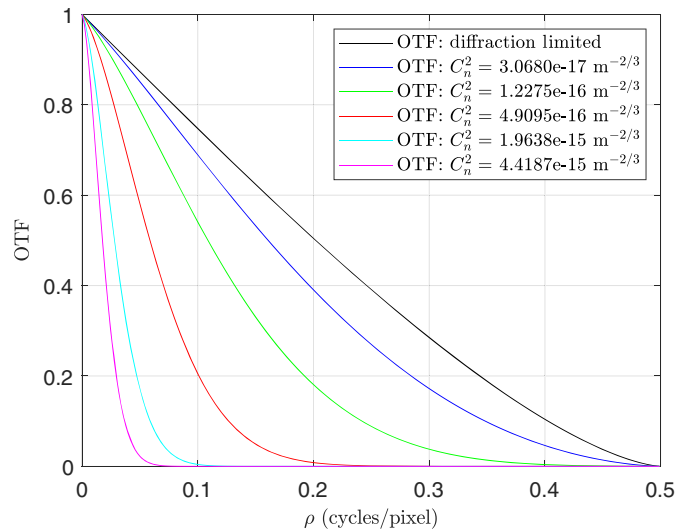


Fig. 1 LE OTFs for the parameters in Tables 1 and 2.

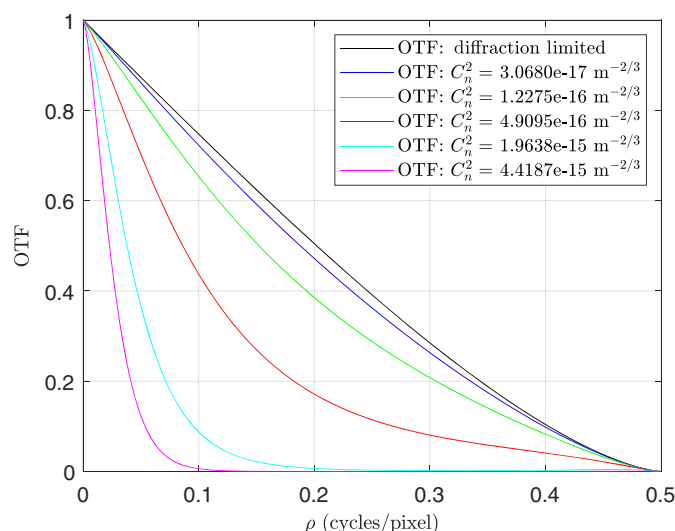


Fig. 2 SE OTFs for the parameters in Tables 1 and 2.

2.2 Anisoplanatic Numerical Wave-Propagation Simulation

The anisoplanatic simulator used here is a numerical wave-propagation method that produces realistic anisoplanatic turbulence degradation. Extensive validation has been performed with this simulator.² Thus, we treat this type of numerical wave simulation method as the gold standard for anisoplanatic data generation. All of the wave-propagation simulation parameters used in this paper are the same as those reported in Hardie et al.² The wave-propagation simulator takes as input the optical parameters of the camera system and the $C_n^2(z)$ profile. It then produces and applies a distinct PSF for every pixel in an input image to produce a degraded frame. It is worth noting that this simulator does not assume an LSI PSF.

The PSFs are formed by propagating point sources from the object plane to the pupil plane through a series of extended phase screens. The basic propagation geometry is shown in Fig. 3. It is worth noting that the dimensions and number of phase screens shown are just for illustrative purposes and are not necessarily the same as those used in Table 1. The left side of the figure

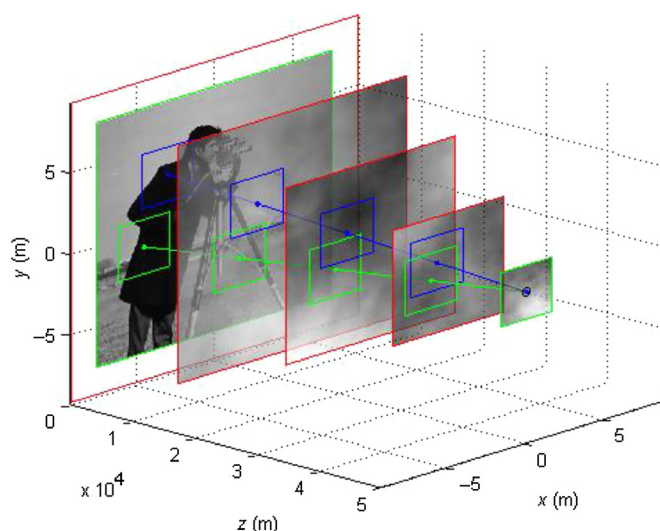


Fig. 3 Anisoplanatic numerical wave-propagation simulation geometry. Point sources in the object plane ($z = 0$) are propagated using numerical wave propagation to the pupil plane of the camera through a series of phase screens. A unique PSF is formed in the focal plane of the simulated camera for each point source in the object plane.

corresponds to the object plane at $z = 0$. The smallest square on the right represents the pupil plane, and the circle within represents the camera aperture. The red boxes represent extended phase screens. We employ 10 screens in our simulations here.² The statistics of the phase screens are based on a modified von Kármán phase power spectral density (PSD).^{2,40} This includes the Kolmogorov PSD as a special case. For each point source in the object plane, the portions of the phase screens along the optical path are extracted and used for numerical wave propagation. These cropped portions are shown as green rectangles for one point source, and blue for another.

Once the PSFs are generated, they are applied in a spatially varying convolution process to produce a degraded image. The centroids of the PSFs will vary and this creates warping. The width of PSFs themselves creates blurring. Because the PSFs vary spatially, the simulator produces anisoplanatic warp and blur. Note that realistic spatial correlation is naturally created because point sources will propagate through overlapping regions of the phase screens as shown in Fig. 3. The closer the point sources, the more overlap the two propagating waves have within the turbulent medium and the more correlation there will be between the resulting PSFs.

2.3 Fast Warping Simulation

The fast warping simulator used here is based on that presented by Schwartzman et al.²⁷ This simulator works by creating 2D arrays of white Gaussian random noise. The noise is filtered with specially designed LSI filters to transform the white noise into random fields with realistic spatial tilt correlation statistics. These shifts are applied to the truth image through interpolation creating realistic geometric distortion.²⁷

There are two important novel aspects of our fast warping simulator that differ from the one proposed by Schwartzman et al.²⁷ First, in addition to the warping we also introduce an LSI PSF blurring using the average SE PSF from Eqs. (5) and (6) with $\alpha = 1$. The other difference is that we employ the tilt statistics derived by Bose-Pillai.⁴⁸ These statistics have been shown to provide excellent agreement with the anisoplanatic simulator statistics.²

It should be noted that the warping generated by our fast simulator is anisoplanatic, but the blurring is isoplanatic. Based on the small isoplanatic angles in Table 2, it is clear that this is not as accurate as the anisoplanatic simulator with respect to individual frames. However, when considering the average of many frames, the spatially varying PSFs will effectively combine to be nearly LSI. Thus, after frame averaging, our fast simulator and the anisoplanatic simulator will tend to be very comparable. We use this to our advantage to rapidly generate suitable training data for our proposed machine learning TM algorithm using the fast simulator. However, if sufficient computational resources are available, one could use the full anisoplanatic simulator to generate training data.

2.4 Simulator Comparison

While the numerical wave anisoplanatic simulator is considered to be very accurate, it does come with a relatively high computational burden. The warping simulator on the other hand, with LSI blurring, is much faster. This makes it much more practical for generating the vast quantities of data necessary for DL training. To compare the speeds of the warping simulator and the anisoplanatic simulator, 30 frames of size 256×256 were generated with both simulators using MATLABTM and the processing times on the same workstation were recorded. The anisoplanatic wave-propagation simulator took 289 s, or nearly 5 min, to complete this task. The warping simulator did the same in just over 3 s. This is ~ 2 orders of magnitude speed enhancement. This timing study has been conducted using a workstation with Intel(R) Core(TM) i9-9980XE CPU @ 3 GHz with 128 GB of RAM and two NVIDIA Titan RTX GPUs. Considering that DL training requires thousands of images, some of which are much larger than 256×256 , one can clearly see the practical benefit of the fast simulator.

However, speed is not the only thing to consider. The fast simulator is only useful if it generates images that are statistically representative of the real image to be processed. In this paper, we focus on TM methods that process an average of registered SE frames. Thus, the simple LSI PSF used by the fast simulator is not problematic since the average of the anisoplanatic images should be similar. This similarity allows use to train of the proposed BM-CNN TM method using

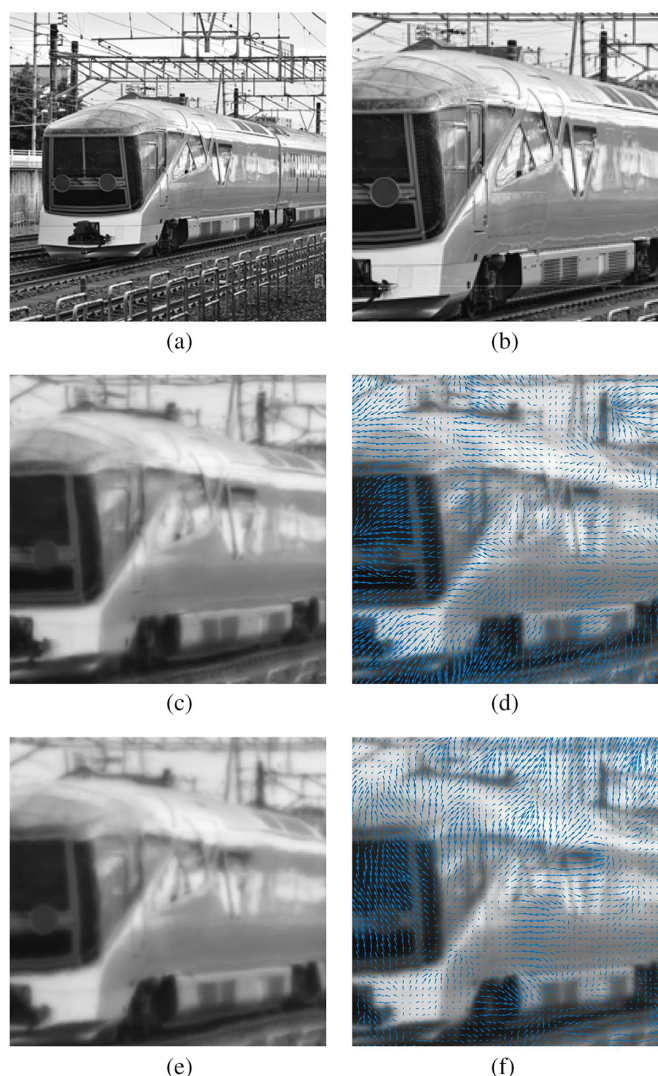


Fig. 4 Turbulence simulator output comparison for level 3 turbulence and no noise. (a) Full size ideal image; (b) 200×200 region of interest; (c) wave-propagation simulation output; (d) wave-propagation output with optical flow; (e) fast warping simulator output; (f) fast warping simulator output with optical flow.

the fast warping simulator. However, for thoroughness and rigor we test using both the fast simulator and the anisoplanatic wave-propagation simulator. As will be seen in Sec. 4, the results show that training with the fast simulator is highly effective.

An example simulated SE frame using both the wave-propagation simulation and the fast warping simulator is shown in Fig. 4. The full ideal image is shown in Fig. 4(a). A 200×200 region of interest is shown in Fig. 4(b). The numerical wave-propagation simulation output for level 3 turbulence is shown in Fig. 4(c). The same image with the optical flow arrows from turbulence warping is shown in Fig. 4(d). Similar results using the fast warping simulator with average SE PSF blur are shown in Figs. 4(e) and 4(f). One can see that the degraded images using the two different simulation methods have a similar level of blur and warping.

3 Turbulence Mitigation

In this section, we define the TM methods studied here. For the reader's convenience, all the methods are listed along with their acronym in Table 3. The raw SE image from the simulator plus noise is referred to simply as SE. The LE image formed as the average of the SE frames is listed as LE in Table 3. The method of forming the average of images registered using BM is

Table 3 TM methods used and their acronyms.

TM method	Acronym
Short exposure	SE
Long exposure	LE
Block matching and averaging	BM-AVG
Long-exposure and Wiener filter	LE-WF
Block matching and Wiener filter	BM-WF
Block matching and convolutional neural network	BM-CNN

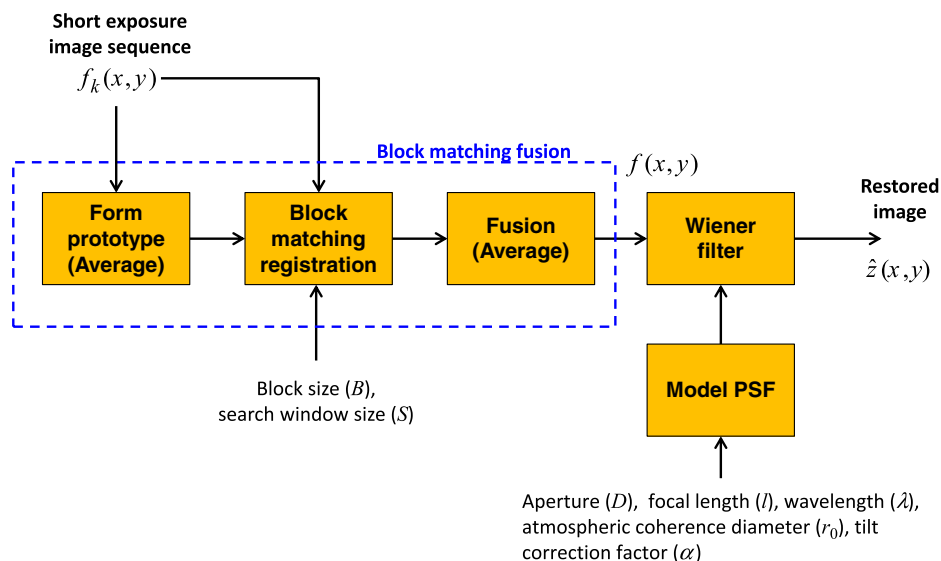
listed as BM-AVG. The next benchmark method listed in Table 3 is the LE followed by restoration using a WF (LE-WF). The theoretical LE PSF is used for the WF and the noise-to-signal (NSR) ratio parameter is optimized on the training data.¹ The last two methods listed in Table 3 combine BM fusion with some type of image restoration. These two methods are described in more detail in the following sections.

3.1 Block Matching and Wiener Filtering

The BM-WF method was proposed and described in detail by Hardie et al in Ref. 1. A block diagram showing the key steps of this method is provided in Fig. 5. The SE frames are acquired and then averaged to form a prototype image with approximately the correct geometry. BM is then used to dewarp the SE images to the greatest extent possible. A $B \times B$ block size is used with $B = 15$ here. The search window is $S \times S$, where $S = 7, 9, 11$ and 13 for the four turbulence levels in Table 2, respectively. The partially dewarped SE images are then averaged. Finally, the average image is processed with a WF. The WF has two parameters, the NSR and the parameter α from Eq. (6) that represents the amount of tilt correction provided by the BM registration step.¹ Here, we optimize these two parameters using the training data to make a fair comparison with the DL method that has access to training data.

3.2 Block Matching and Convolutional Neural Network

In the proposed BM-CNN method, we use a CNN instead of the WF to perform restoration. A block diagram detailing the process can be seen in Fig. 6. The CNN is non-linear in nature

**Fig. 5** BM-WF TM block diagram.¹

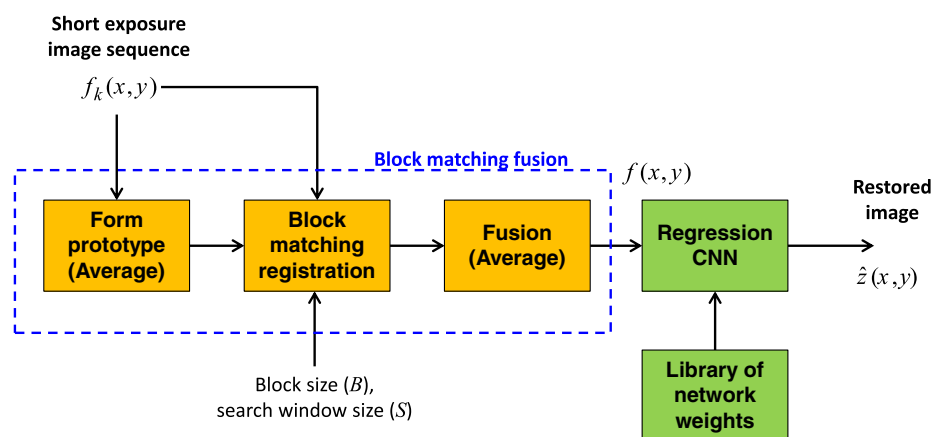


Fig. 6 Proposed BM-CNN TM block diagram. It is worth noting that we have replaced the WF in the BM-WF method with a CNN to improve performance.

and we believe it is better equipped to address the non-stationary nature of the imagery that results from the BM fusion step.

As previously stated in Sec. 1, our BM-CNN uses the FIFNet architecture. In particular, 10 convolution (Conv) + rectified linear unit (ReLU) activation layers are employed. The first five convolution layers use a filter size of 3×3 , and the last five use a filter size of 5×5 . A diagram of the network architecture is shown in Fig. 7. A listing and description of all of the layers is

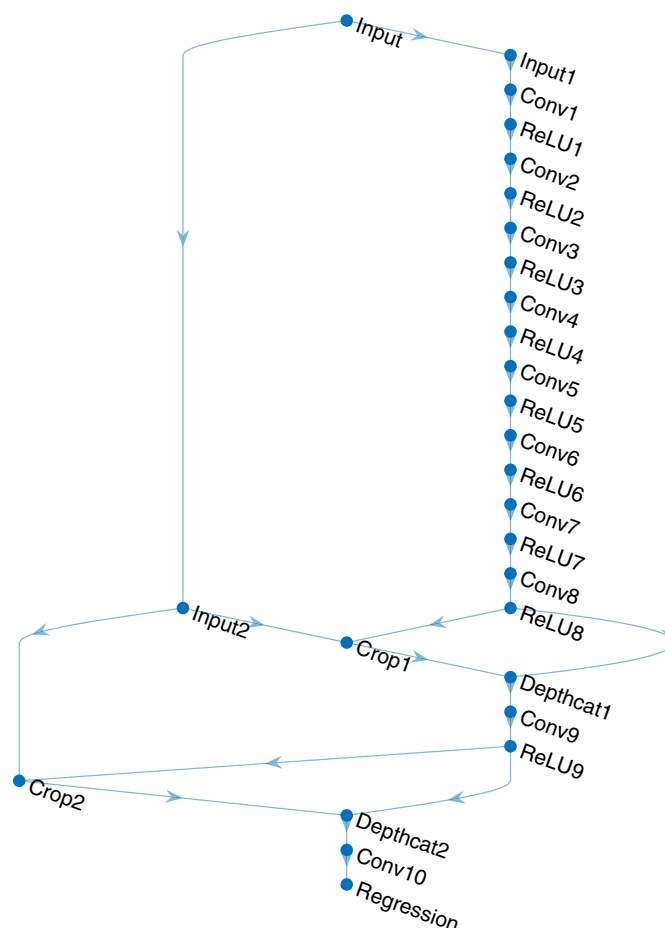


Fig. 7 FIFNet CNN network architecture used for the proposed BM-CNN TM method.

Table 4 Description of the FIFNet layers³⁸ used for the proposed BM-CNN TM method. The spatial dimensions of the activations are for training using 61×61 image patches.

Layers	Type	Spatial kernel	Activations
1	Input	—	$61 \times 61 \times 1$
2	Splitting	—	$61 \times 61 \times 1$
3, 4	2D conv+ReLU	3×3	$59 \times 59 \times 64$
5, 6	2D conv+ReLU	3×3	$57 \times 57 \times 64$
7, 8	2D conv+ReLU	3×3	$55 \times 55 \times 64$
9, 10	2D conv+ReLU	3×3	$53 \times 53 \times 64$
11, 12	2D conv+ReLU	3×3	$51 \times 51 \times 64$
13, 14	2D Conv+ReLU	5×5	$47 \times 47 \times 64$
15, 16	2D conv+ReLU	5×5	$43 \times 43 \times 64$
17, 18	2D conv+ReLU	5×5	$39 \times 39 \times 64$
19, 20	2D crop+ReLU	—	$39 \times 39 \times 1$
21	Concatenate	—	$39 \times 39 \times 65$
22, 23	2D conv+ReLU	5×5	$35 \times 35 \times 65$
24	2D ccrop	—	$35 \times 35 \times 1$
25	Concatenate	—	$35 \times 35 \times 66$
26	2D conv	5×5	$31 \times 31 \times 1$
27	Regression	—	—

provided in Table 4. As can be seen in Table 4, the total number of layers is 27. By current standards, this represents a relatively small network. This allows the network to be trained from scratch in a relatively short time. The network is implemented in MATLAB™ using the Deep Learning Toolbox™.

Training and testing of the network are done here using the DIV2K database.⁴⁹ This dataset has become popular for testing image restoration and image SR methods. It contains 800 high-resolution images for training and 100 for testing. The original images are 24-bit RGB images with 2K resolution. All of the input images have been converted to grayscale with a floating point dynamic range of 0 to 1. They have also been reduced in size by a scaling factor of 0.25.

The basic training process is shown in Fig. 8. The undegraded training data serve as the “truth” images. These images are fed into the fast warping simulator to produce the degraded images. BM is applied to these degraded images and then the registered images are then averaged and fed into the CNN. During training iterations, image patches of size 61×61 are extracted from random positions in each training image. The corresponding truth patches of size 31×31 are also extracted for regression. The convolutional layers of the CNN do not use zero padding, so the processed image patches get smaller with each layer as shown in Table 4. This is why the truth patch is smaller than the input patch. The advantage of this is that it allows for a smaller network and avoids extrapolation errors on the patch borders.³⁸

CNN updates during training are done using stochastic gradient descent with momentum optimization. Sixty four different random patches are extracted from each training image. We use a mini-batch size of 64 so that each iteration corresponds to one training image. All training is done with a total of 100 epochs. We set the initial learning rate to 0.1 and reduced this by a factor of 0.1 after every 10 epochs. Using the same workstation described in Sec. 2.4, the total training time for this network is ~ 100 min.

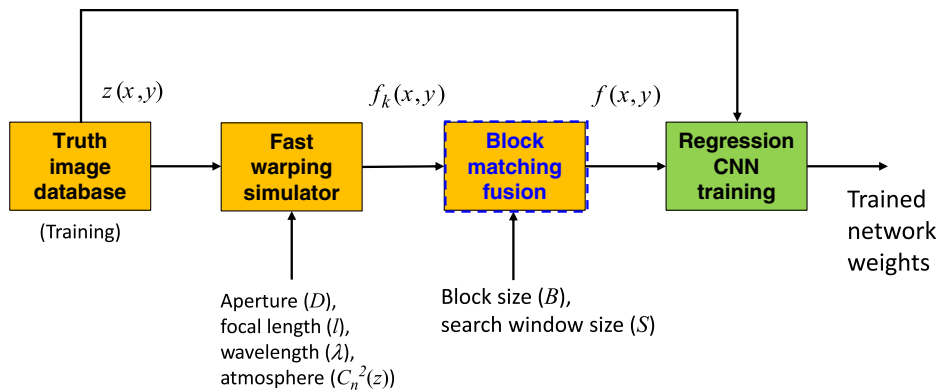


Fig. 8 BM-CNN training process block diagram.

4 Experimental Results

The experimental results have all been conducted using the optical parameters from Table 1 and the turbulence degradation parameters from Table 2. The four different turbulence levels have all been trained using the fast warping simulator for noise standard deviation of 10 and 30, relative to an 8-bit grayscale image range. The results are all formed using image sequences of 30 SE input frames for each turbulence and noise scenario generated from each of the original DIV2K 800 ideal truth images. For testing, the 100 DIV2K validation images are used. Here, we generate the 30 frame input sequences using both the fast warping simulator and the anisoplanatic simulator.

4.1 Quantitative Results

To quantitatively compare results of the all TM methods, the metrics of peak signal-to-noise ratio (PSNR) and structural similarity index measure of image quality (SSIM)⁵⁰ are used. In both metrics, the higher the number the better. For SSIM the numbers range from 0 to 1, so getting numbers close to 1 is desired. The quantitative error metric values are averaged over the 100 validation images with ignore border of 5 pixels on all sides to exclude any border artifacts.

The PSNR results are provided in Tables 5–8 for the four turbulence levels in Table 2. The corresponding SSIM results are in Tables 9–12. Data generated from the fast warping

Table 5 Average PSNR (dB) results for 100 validation sequences with turbulence level 1 and 30 input frames.

Mitigation	Simulated data: training/testing			
	Noise std = 10		Noise std = 30	
	Warp/Warp	Warp/Aniso	Warp/Warp	Warp/Aniso
SE	23.3489	23.2323	17.6345	17.6083
LE	26.1827	25.9293	25.3834	25.1803
BM-AVG	26.7063	26.4955	25.4702	25.3453
LE-WF	29.7577	29.7500	27.2442	27.1366
BM-WF	30.3766	30.3271	27.2241	27.2097
BM-CNN	31.8466	31.6640	28.8948	28.7915

Table 6 Average PSNR (dB) results for 100 validation sequences with turbulence level 2 and 30 input frames.

Mitigation	Simulated data: training/testing			
	Noise std = 10		Noise std = 30	
	Warp/Warp	Warp/Aniso	Warp/Warp	Warp/Aniso
SE	21.6393	21.4881	17.0607	17.0140
LE	24.1923	23.8974	23.6663	23.4118
BM-AVG	25.6641	25.5496	24.4865	24.4400
LE-WF	27.1175	26.9127	25.5973	25.3992
BM-WF	29.2104	29.3532	26.3433	26.4179
BM-CNN	30.2577	30.3175	27.5861	27.6535

Table 7 Average PSNR (dB) results for 100 validation sequences with turbulence level 3 and 30 input frames.

Mitigation	Simulated data: training/testing			
	Noise std = 10		Noise std = 30	
	Warp/Warp	Warp/Aniso	Warp/Warp	Warp/Aniso
SE	19.7836	19.5549	16.2892	16.1915
LE	21.7032	21.3931	21.4016	21.1164
BM-AVG	23.2272	23.1571	22.3702	22.3104
LE-WF	23.8516	23.4875	23.2291	22.9434
BM-WF	26.2424	26.5105	24.1303	24.2210
BM-CNN	26.5992	26.8513	24.8633	24.9602

Table 8 Average PSNR (dB) results for 100 validation sequences with turbulence level 4 and 30 input frames.

Mitigation	Simulated data: training/testing			
	Noise std = 10		Noise std = 30	
	Warp/Warp	Warp/Aniso	Warp/Warp	Warp/Aniso
SE	18.2356	17.9519	15.5183	15.3691
LE	19.5227	19.2212	19.3411	19.0539
BM-AVG	20.2898	20.1493	19.8932	19.7544
LE-WF	21.2001	20.7712	20.9670	20.6208
BM-WF	22.4137	22.3624	21.4067	21.3097
BM-CNN	22.5889	22.5473	21.6959	21.6129

Table 9 Average SSIM results for 100 validation sequences with turbulence level 1 and 30 input frames.

Mitigation	Simulated data: training/testing			
	Noise std = 10		Noise std = 30	
	Warp/Warp	Warp/Aniso	Warp/Warp	Warp/Aniso
SE	0.5435	0.5576	0.2482	0.2627
LE	0.8060	0.8055	0.7005	0.7064
BM-AVG	0.8230	0.8250	0.6856	0.6969
LE-WF	0.8572	0.8652	0.7607	0.7701
BM-WF	0.8630	0.8717	0.7444	0.7585
BM-CNN	0.9264	0.9299	0.8556	0.8613

Table 10 Average SSIM results for 100 validation sequences with turbulence level 2 and 30 input frames.

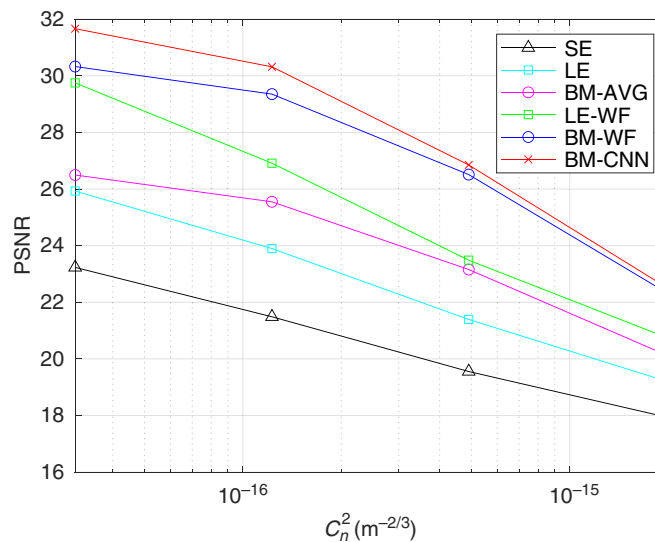
Mitigation	Simulated data: training/testing			
	Noise Std = 10		Noise Std = 30	
	Warp/Warp	Warp/Aniso	Warp/Warp	Warp/Aniso
SE	0.4589	0.4702	0.2071	0.2197
LE	0.7086	0.7042	0.6088	0.6108
BM-AVG	0.7817	0.7887	0.6350	0.6515
LE-WF	0.7960	0.8009	0.7065	0.7125
BM-WF	0.8395	0.8542	0.7181	0.7373
BM-CNN	0.9006	0.9104	0.8158	0.8287

Table 11 Average SSIM results for 100 validation sequences with turbulence level 3 and 30 input frames.

Mitigation	Simulated data: training/testing			
	Noise std = 10		Noise std = 30	
	Warp/Warp	Warp/Aniso	Warp/Warp	Warp/Aniso
SE	0.3647	0.3697	0.1622	0.1706
LE	0.5597	0.5489	0.4733	0.4687
BM-AVG	0.6534	0.6644	0.5160	0.5296
LE-WF	0.6712	0.6673	0.6077	0.6067
BM-WF	0.7482	0.7772	0.6277	0.6493
BM-CNN	0.7936	0.8180	0.7095	0.7275

Table 12 Average SSIM results for 100 validation sequences with turbulence level 4 and 30 input frames.

Mitigation	Simulated Data: training/testing			
	Noise std = 10		Noise std = 30	
	Warp/Warp	Warp/Aniso	Warp/Warp	Warp/Aniso
SE	0.2963	0.2973	0.1324	0.1376
LE	0.4618	0.4458	0.3870	0.3771
BM-AVG	0.4903	0.4839	0.3992	0.3966
LE-WF	0.5417	0.5283	0.5186	0.5078
BM-WF	0.5788	0.5911	0.5228	0.5251
BM-CNN	0.6033	0.6168	0.5540	0.5584

**Fig. 9** PSNR (dB) results for the one hundred 30-frame validation sequences for the turbulence levels in Table 2 and a noise standard deviation of 10.

simulator are indicated with the designation “Warp” in the results tables. Data from the anisoplanatic numerical wave-propagation simulator are designated “Aniso.” Bold entries in the tables correspond to the best results. For the reader’s convenience, the quantitative PSNR values are also plotted in Figs. 9 and 10 for noise standard deviations of 10 and 30, respectively. The corresponding SSIM values are plotted in Figs. 11 and 12. The plotted data are for testing on the numerical wave-propagation anisoplanatic simulator.

One can see that the raw SE images score the lowest in the performance metrics. The LE images perform better. This is due to the geometric correction provided by the frame averaging. Averaging the frames after BM with the BM-AVG method provides further improvement. Here we get the benefit of the LE geometric correction but with less turbulence motion blur. The LE-WF gives the next best result as it includes restoration from the WF. Then we have the BM-WF improving upon that, thanks to both the registration and restoration. Finally, we see that the BM-CNN gives the highest quantitative performance values. The relative improvement is most pronounced at lower turbulence levels and higher noise levels.

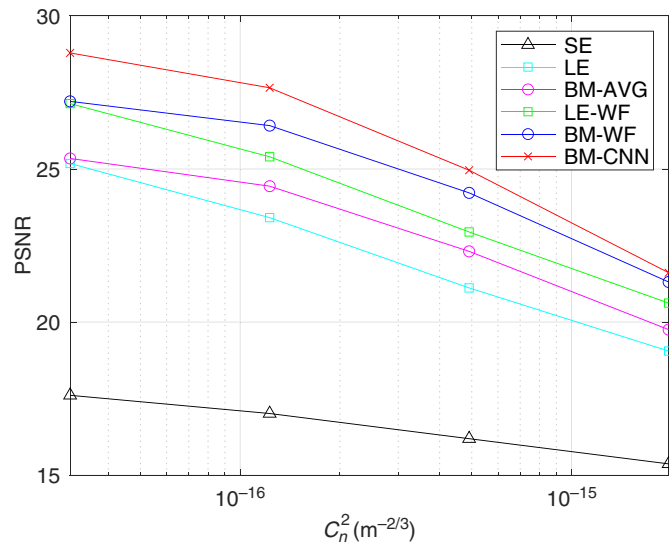


Fig. 10 PSNR (dB) results for the one hundred 30-frame validation sequences for the turbulence levels in Table 2 and a noise standard deviation of 30.

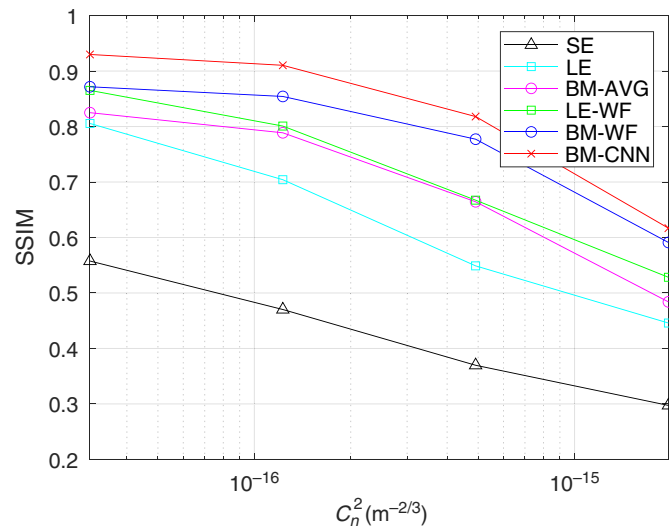


Fig. 11 SSIM results for the one hundred 30-frame validation sequences for the turbulence levels in Table 2 and a noise standard deviation of 10.

4.2 Qualitative Image Results

Image results are shown in Figs. 13–16 for subjective evaluation. All of the images are 200×200 regions of interest with the same layout. Image (a) is the truth image and (b) is the raw SE image. The LE and LE-WF are shown in (c) and (d), respectively. The BM-WF is shown in (e), and the the BM-CNN is shown in (f).

Figure 13 shows the penguin image for level 2 turbulence noise standard deviation of 30. The high noise level is evident in the raw SE image in Fig. 13(b). The LE image in Fig. 13(c) has reduced noise from averaging 30 frames but is highly blurred. The sharpening effect of the WF in Fig. 13(d) is clear in the LE-WF image. The BM-WF in Fig. 13(e) is sharper than the LE-WF because of the reduced turbulence motion blurring from the registration. Finally, note that the BM-CNN image in Fig. 13(f) appears to have the same level of sharpness as the BM-WF, but with superior noise suppression. The improved noise suppression is clearly visible in the white

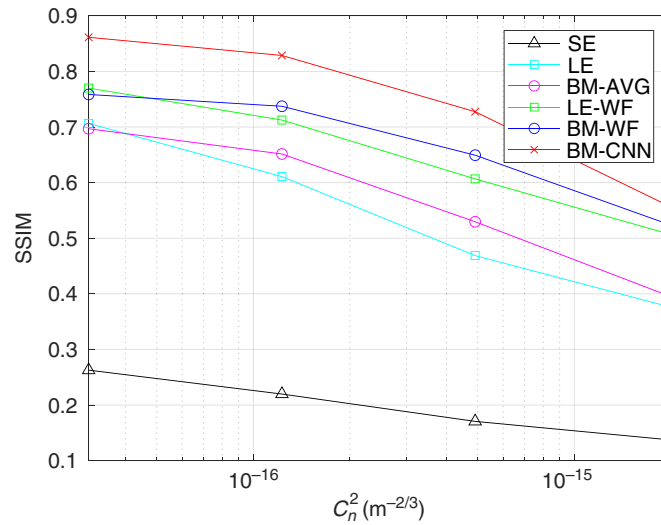


Fig. 12 SSIM results for the one hundred 30-frame validation sequences for the turbulence levels in Table 2 and a noise standard deviation of 30.

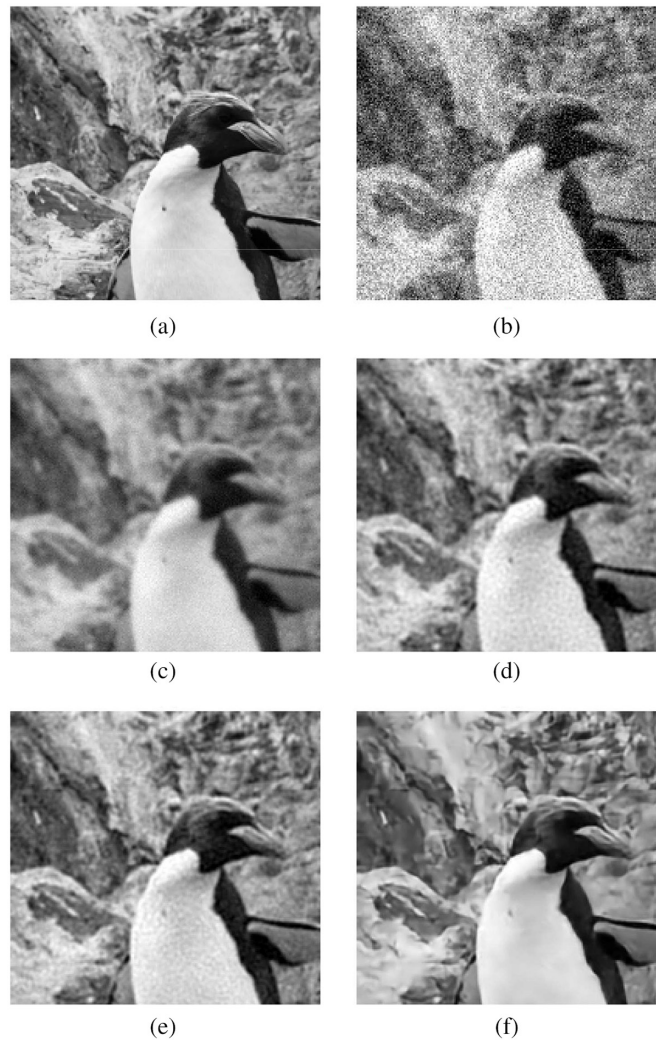


Fig. 13 TM results using the anisoplanatic simulator data as testing data (Penguin). These results are for turbulence level 2, 30 input frames, and noise standard deviation of 30. (a) Truth; (b) SE; (c) LE; (d) LE-WF; (e) BM-WF; (f) BM-CNN.

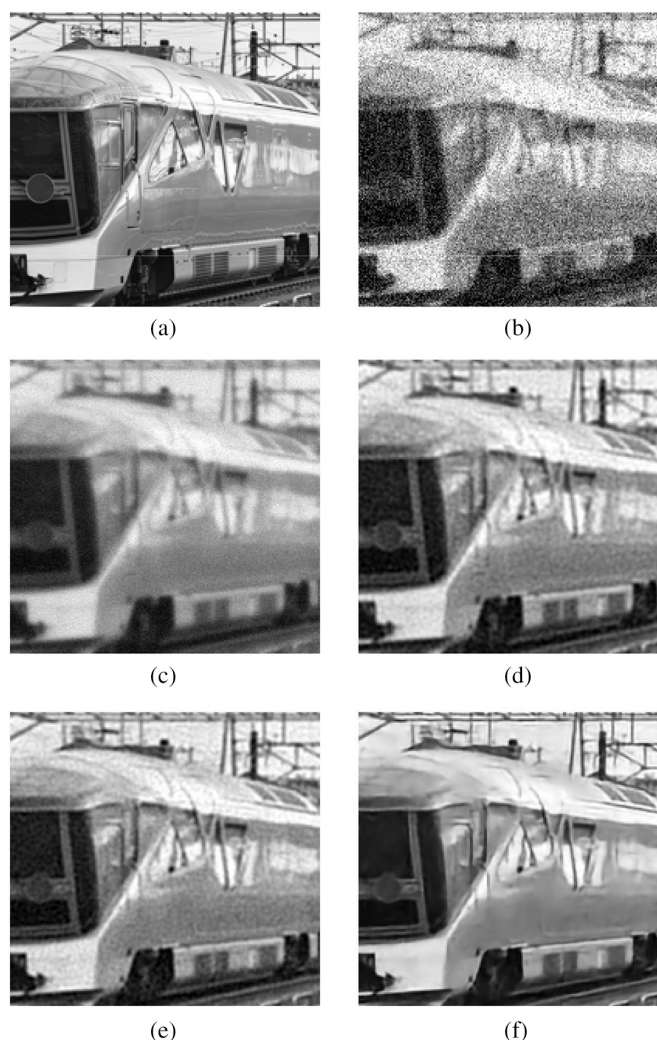


Fig. 14 TM results using the anisoplanatic simulator data as testing data (Train). These results are for turbulence level 2 and 30 input frames, and noise standard deviation of 30. (a) Truth; (b) SE; (c) LE; (d) LE-WF; (e) BM-WF; (f) BM-CNN.

region of the penguin's belly. We attribute this to the ability of the CNN to be non-linear and spatially adaptive. In contrast, the fixed WF is limited to LSI processing. The same scenario is shown in Fig. 14 for an image of a train. Similar relative performance of the methods is observed here as well.

Level 3 turbulence results for the penguin image are shown in Fig. 15 with a lower noise standard deviation of 10. The lower noise is evident by comparing Fig. 15(b) to Fig. 13(b). The increase in turbulence blurring is most pronounced when comparing Figs. 15(c) to Fig. 13(c). Here, both the BM-WF and BM-CNN provide good results, but again the BM-CNN provides the same level of sharpness with noticeably more noise reduction. The same scenario for the image of a train is shown in Fig. 16 with similar subjective results.

5 Conclusions

A DL approach to combating the image degradation from atmospheric turbulence is introduced in this paper. In particular, we proposed the BM-CNN TM method that builds on earlier work by some of the current authors with the BM-WF¹ method. In our new approach, we have replaced the WF with a CNN and employ DL to train the network. The CNN architecture we use here is based on the FIFNet.³⁸ The FIFNet has a lightweight architecture with relatively few layers.

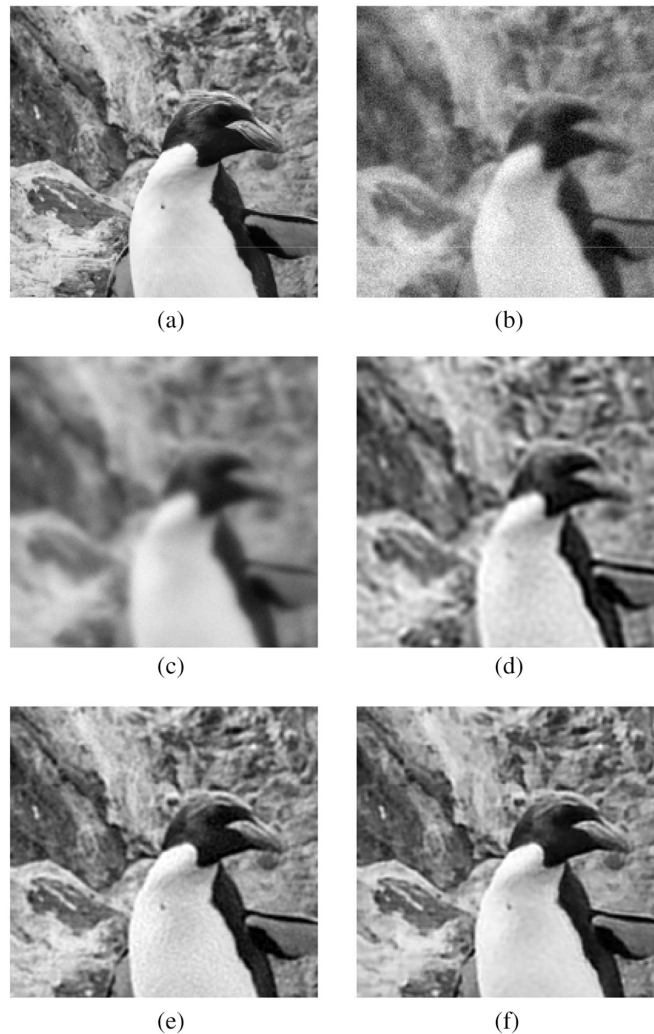


Fig. 15 TM results using the anisoplanatic simulator data as testing data (Penguin). These results are for turbulence level 3, 30 input frames, and noise standard deviation of 10. (a) Truth; (b) SE; (c) LE; (d) LE-WF; (e) BM-WF; (f) BM-CNN.

This allows us to train the network from scratch, without the need for transfer learning, in ~ 100 min. The BM-CNN method is one of the first in the literature to apply a CNN to the problem of TM.

In addition to the important innovation of using a CNN for TM, we have also employed two different turbulence simulation methods for the first time. Because of the high data requirements for training, we employ a fast warping simulator for generating training images. The fast simulator extends the method introduced by Schwartzman et al.²⁷ Our novel modifications include the use of different tilt statistics and the introduction of LSI blurring using the average SE PSF, as described in Sec. 2. While training is done exclusively with the fast simulator, testing is performed here using both the fast simulator, and a numerical wave-propagation anisoplanatic simulator.² We believe numerical wave-propagation remains the gold standard for anisoplanatic turbulence simulation. However, the fast simulator is ~ 100 times faster. The BM-CNN trained on the fast warping simulator gives almost identical performance results on testing data generated by both the warping simulator and the numerical wave-propagation anisoplanatic simulator. We believe this shows that the computationally faster simulation method is well suited for training. We believe it is very significant that we have demonstrated that the fast simulator can be effectively used for training in this application. The use of the fast simulator dramatically increases the speed of training data generation and makes the DL process much more practical in this application.

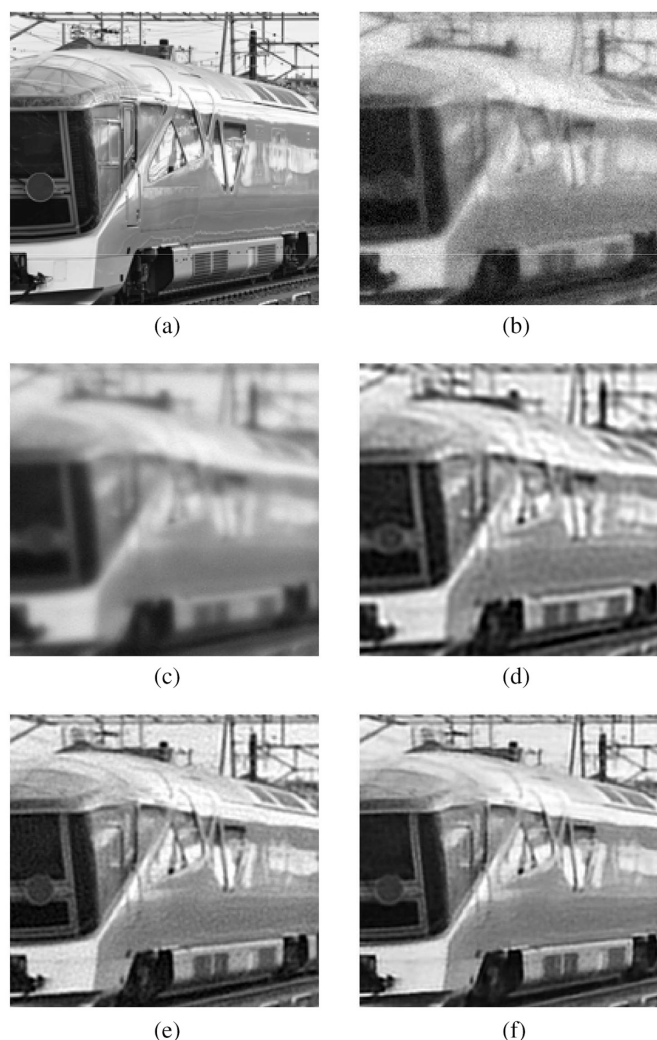


Fig. 16 TM results using the anisoplanatic simulator data as testing data (Train). These results are for turbulence level 3 and 30 input frames, and noise standard deviation of 10. (a) Truth; (b) SE; (c) LE; (d) LE-WF; (e) BM-WF; (f) BM-CNN.

The proposed BM-CNN is compared to several benchmark methods in Sec. 4. In all of the quantitative performance experiments conducted, the BM-CNN was the best performing method. The BM-WF was the next best performing method. In subjective evaluation of the images, we have observed that the big advantage of the BM-CNN appears to be in how it handles noise. The non-linear and spatially adaptive CNN method outperforms the WF and provides noticeably more noise suppression in the flat areas while providing the same level of sharpness in the dynamic areas. This gives very pleasing visual results that are supported by the favorable performance metrics.

Acknowledgments

The authors would like to thank Joe French, Bruce Wilcoxon, and the project team at Leidos for project management support. This work was supported by the Air Force Research Laboratory (AFRL), Sensors Directorate under contract FA8650-18-F-1710. The contents of this document are approved for public release under case number AFRL 88ABW-2020-3547. The authors declare that there are no conflicts of interest to disclose.

References

1. R. C. Hardie et al., "Block matching and Wiener filtering approach to optical turbulence mitigation and its application to simulated and real imagery with quantitative error analysis," *Opt. Eng.* **56**(7), 071503 (2017).
2. R. C. Hardie et al., "Simulation of anisoplanatic imaging through optical turbulence using numerical wave propagation with new validation analysis," *Opt. Eng.* **56**(7), 071502 (2017).
3. M. C. Roggemann and B. M. Welsh, *Imaging Through Turbulence*, *Laser and Optical Science and Technology*, CRC Press (1996).
4. R. E. Hufnagel and N. R. Stanley, "Modulation transfer function associated with image transmission through turbulent media," *J. Opt. Soc. Am.* **54**, 52–61 (1964).
5. A. W. M. van Eekeren et al., "Turbulence compensation: an overview," *Proc. SPIE* **8355**, 83550Q (2012).
6. A. Lambert et al., "Experiments in turbulence induced super-resolution in surveillance imagery," *Proc. SPIE* **7126**, 712612 (2009).
7. Y. Liu, "Mitigation of the effects of turbulence in the imaging path," PhD thesis, University of New South Wales Australian Defence Force Academy (2016).
8. C. L. Matson et al., "Multiframe blind deconvolution and bispectrum processing of atmospherically degraded data: a comparison," *Proc. SPIE* **4792**, 55–66 (2002).
9. G. E. Archer, J. P. Bos, and M. C. Roggemann, "Reconstruction of long horizontal-path images under anisoplanatic conditions using multiframe blind deconvolution," *Opt. Eng.* **52**(8), 083108 (2013).
10. G. E. Archer, J. P. Bos, and M. C. Roggemann, "Comparison of bispectrum, multiframe blind deconvolution and hybrid bispectrum-multiframe blind deconvolution image reconstruction techniques for anisoplanatic, long horizontal-path imaging," *Opt. Eng.* **53**(4), 043109 (2014).
11. D. Fraser, G. Thorpe, and A. Lambert, "Atmospheric turbulence visualization with wide-area motion-blur restoration," *J. Opt. Soc. Am. A* **16**, 1751–1758 (1999).
12. D. H. Frakes, J. W. Monaco, and M. J. T. Smith, "Suppression of atmospheric turbulence in video using an adaptive control grid interpolation approach," *IEEE Int. Conf. Acoustics, Speech, and Signal Process., 2001. Proc.*, Vol. 3, pp. 1881–1884 (2001).
13. D. Li, R. M. Mersereau, and S. Simske, "Atmospheric turbulence-degraded image restoration using principal components analysis," *IEEE Geosci. Remote Sens. Lett.* **4**, 340–344 (2007).
14. X. Zhu and P. Milanfar, "Image reconstruction from videos distorted by atmospheric turbulence," *Proc. SPIE* **7543**, 75430S (2010).
15. X. Zhu and P. Milanfar, "Removing atmospheric turbulence via space-invariant deconvolution," *IEEE Trans. Pattern Anal. Mach. Intell.* **35**, 157–170 (2013).
16. T. W. Lawrence et al., "Experimental validation of extended image reconstruction using bispectral speckle interferometry," *Proc. SPIE* **1237**, 522–537 (1990).
17. T. W. Lawrence et al., "Extended-image reconstruction through horizontal path turbulence using bispectral speckle interferometry," *Opt. Eng.* **31**(3), 627–636 (1992).
18. C. J. Carrano, "Speckle imaging over horizontal paths," *Proc. SPIE* **4825**, 109–120 (2002).
19. C. J. Carrano, "Anisoplanatic performance of horizontal-path speckle imaging," *Proc. SPIE* **5162**, 14–27 (2003).
20. J. P. Bos and M. C. Roggemann, "Mean squared error performance of speckle-imaging using the bispectrum in horizontal imaging applications," *Proc. SPIE* **8056**, 805603 (2011).
21. J. P. Bos and M. C. Roggeman, "The effect of free parameter estimates on the reconstruction of turbulence corrupted images using the bispectrum," *Proc. SPIE* **8161**, 816105 (2011).
22. J. P. Bos and M. C. Roggemann, "Robustness of speckle-imaging techniques applied to horizontal imaging scenarios," *Opt. Eng.* **51**(8), 083201 (2012).
23. J. P. Bos and M. C. Roggemann, "Blind image quality metrics for optimal speckle image reconstruction in horizontal imaging scenarios," *Opt. Eng.* **51**(10), 107003 (2012).
24. J. P. Bos and M. C. Roggeman, "Simulation of extended scenes imaged through turbulence over horizontal paths," *Proc. SPIE* **8161**, 816106 (2011).

25. J. P. Bos and M. C. Roggemann, "Technique for simulating anisoplanatic image formation over long horizontal paths," *Opt. Eng.* **51**, 101704 (2012).
26. G. Monnier, F.-R. Duval, and S. Amram, "GPU-based simulation of optical propagation through turbulence for active and passive imaging," *Proc. SPIE* **9242**, 92421R (2014).
27. A. Schwartzman et al., "Turbulence-induced 2d correlated image distortion," in *IEEE Int. Conf. Comput. Photogr. (ICCP)*, Los Alamitos, California, IEEE Computer Society, pp. 1–13 (2017).
28. V. Jain et al., "Supervised learning of image restoration with convolutional networks," in *IEEE 11th Int. Conf. Comput. Vision*, pp. 1–8 (2007).
29. K. Zhang et al., "Learning deep cnn denoiser prior for image restoration," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.* (2017).
30. X. Mao, C. Shen, and Y.-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," in *Adv. Neural Inf. Process. Syst.* 29, D. D. Lee et al., Eds., Curran Associates, Inc., pp. 2802–2810 (2016).
31. Y. Chang et al., "HSI-DeNet: hyperspectral image restoration via convolutional neural network," *IEEE Trans. Geosci. Remote Sens.* **57**(2), 667–682 (2019).
32. W. Dong et al., "Denoising prior driven deep neural network for image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.* **41**(10), 2305–2318 (2019).
33. C. Dong et al., "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.* **38**(2), 295–307 (2016).
34. X. Ji, Y. Lu, and L. Guo, "Image super-resolution with deep convolutional neural network," in *IEEE First Int. Conf. Data Sci. in Cybersp.*, pp. 626–630 (2016).
35. M. Kawulok et al., "Deep learning for multiple-image super-resolution," *IEEE Geosci. Remote Sens. Lett.* **17**(6), 1062–1066 (2020).
36. J. Chen et al., "Single image super-resolution based on deep learning and gradient transformation," in *IEEE 13th Int. Conf. Signal Process.*, pp. 663–667 (2016).
37. J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 1646–1654 (2016).
38. H. Elwarfalli and R. C. Hardie, "Fifnet: a convolutional neural network for motion-based multiframe super-resolution using fusion of interpolated frames," *Comput. Vision Image Understanding* **202**, 103097 (2021).
39. J. Gao, N. Anantrasirichai, and D. Bull, "Atmospheric turbulence removal using convolutional neural network," ArXiv abs/1912.11350 (2019).
40. J. D. Schmidt, *Numerical Simulation of Optical Wave Propagation with Examples in MATLAB*, SPIE Press Monograph, SPIE Press (2010).
41. R. C. Hardie et al., "Fusion of interpolated frames superresolution in the presence of atmospheric optical turbulence," *Opt. Eng.* **58**(8), 083103 (2019).
42. R. K. Tyson, *Adaptive Optics Engineering Handbook*, CRC Press (1999).
43. L.-K. Yu et al., "Isoplanatic angle in finite distance: experimental validation," *Opt. Eng.* **54**(2), 024105 (2015).
44. J. W. Goodman, *Introduction to Fourier Optics*, 3rd ed., Roberts and Company Publishers (2004).
45. N. S. Kopeika, "Effects of aerosols on imaging through the atmosphere: a review of spatial frequency and wavelength dependent effects," *Opt. Eng.* **24**(4), 707–712 (1985).
46. D. A. LeMaster and M. T. Eismann, "Impact of atmospheric aerosols on long range image quality," *Proc. SPIE* **8355**, 83550F (2012).
47. M. T. Eismann and D. A. LeMaster, "Aerosol modulation transfer function model for passive long-range imaging over a nonuniform atmospheric path," *Opt. Eng.* **52**(4), 046201 (2013).
48. S. R. Bose-Pillai et al., "Estimation of atmospheric turbulence using differential motion of extended features in time-lapse imagery," *Opt. Eng.* **57**(10), 104108 (2018).
49. E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: dataset and study," in *IEEE Conf. Comput. Vision and Pattern Recognit. Workshops*, pp. 1122–1131 (2017).
50. Z. Wang et al., "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.* **13**(4), 600–612 (2004).

Matthew A. Hoffmire received his master's degree in electrical engineering from the University of Dayton in December 2020. He worked as a graduate research assistant with the Air Force Research Laboratory, Wright-Patterson AFB, OH, after receiving his BS degree in computer engineering at the University of Dayton in 2018. His research interests include digital signal and image processing and machine learning.

Russell C. Hardie is a full professor in the Department of Electrical and Computer Engineering at the University of Dayton and holds a joint appointment in the Department of Electro-Optics and Photonics. He received the University of Dayton's top university-wide teaching award in 2006 (the Alumni Award in Teaching). He also received the Rudolf Kingslake Medal and Prize from SPIE in 1998 for work on super-resolution imaging. His research interests include a wide range of topics in signal and image processing, image restoration, medical image analysis, and machine learning.

Michael A. Rucci received his BS and MS degrees in electrical engineering from the University of Dayton in 2012 and 2014, respectively. He is a research engineer at the Air Force Research Laboratory, Wright-Patterson AFB, OH, and his current research interests include day/night passive imaging, turbulence modeling and simulation, and image processing. He and three other researchers received the Rudolf Kingslake Medal from SPIE in 2018 for their work in turbulence modeling.

Richard Van Hook is a research engineer at the Air Force Research Laboratory, Wright-Patterson Air Force Base, Ohio. He received his MS and BS degrees in computer engineering from Wright State University in 2014 and 2008, respectively. His research interests include image processing, atmospheric modeling, and TM.

Barry K. Karch is a principal research electronics engineer in the Multispectral Sensing & Detection Division, Sensors Directorate of the Air Force Research Laboratory, Wright-Patterson AFB, OH. He received his BS degree in electrical engineering (1987), MS degree in electro-optics and electrical engineering (1992/1994), and PhD in electrical engineering (2015) from the University of Dayton, Dayton, Ohio. He has worked in the areas of EO/IR remote sensor system and processing development for 30 years.