Optical Engineering

OpticalEngineering.SPIEDigitalLibrary.org

Synthesis of atmospheric turbulence point spread functions by sparse and redundant representations

Bobby R. Hunt
Amber L. Iler
Christopher A. Bailey
Michael A. Rucci

SPIE.

Bobby R. Hunt, Amber L. Iler, Christopher A. Bailey, Michael A. Rucci, "Synthesis of atmospheric turbulence point spread functions by sparse and redundant representations," *Opt. Eng.* **57**(2), 024101 (2018), doi: 10.1117/1.OE.57.2.024101.

Optical Engineering 57(2), 024101 (February 2018)

Synthesis of atmospheric turbulence point spread functions by sparse and redundant representations

Bobby R. Hunt,[a,*] Amber L. Iler,[b,*] Christopher A. Bailey,[c] and Michael A. Rucci[d]
[a]Integrity Applications Incorporated-Pacific Defense Solutions, Kihei, Hawaii, United States
[b]Integrity Applications Incorporated, Ann Arbor, Michigan, United States
[c]Leidos, Beavercreek, Ohio, United States
[d]Air Force Research Laboratory, Wright–Patterson Air Force Base, Ohio, United States

Abstract. Atmospheric turbulence is a fundamental problem in imaging through long slant ranges, horizontal-range paths, or uplooking astronomical cases through the atmosphere. An essential characterization of atmospheric turbulence is the point spread function (PSF). Turbulence images can be simulated to study basic questions, such as image quality and image restoration, by synthesizing PSFs of desired properties. In this paper, we report on a method to synthesize PSFs of atmospheric turbulence. The method uses recent developments in sparse and redundant representations. From a training set of measured atmospheric PSFs, we construct a dictionary of "basis functions" that characterize the atmospheric turbulence PSFs. A PSF can be synthesized from this dictionary by a properly weighted combination of dictionary elements. We disclose an algorithm to synthesize PSFs from the dictionary. The algorithm can synthesize PSFs in three orders of magnitude less computing time than conventional wave optics propagation methods. The resulting PSFs are also shown to be statistically representative of the turbulence conditions that were used to construct the dictionary. [DOI: 10.1117/1.OE.57.2.024101]

Keywords: image modeling and simulation; turbulence; sparse and redundant representations; point spread functions; remote sensing; image restoration.

Paper 171358 received Sep. 6, 2017; accepted for publication Jan. 5, 2018; published online Feb. 7, 2018.

1 Introduction and Background

In any long-range imaging case, turbulence degrades the imagery by inducing both warping and blurring. The effects of turbulence are caused by continual atmospheric changes in pressure, temperature, and turbulent movement, leading to random fluctuations in the index of refraction.[1] Imaging through turbulence is usually separated into isoplanatic and anisoplanatic imaging. The first case is prevalent in astronomical imaging, where one turbulence point spread function (PSF) degrades the entire image, leading to distorted imagery. The anisoplanatic imaging case has spatially varying PSFs leading to variations in blurring and warping across the image plane. Anisoplanatic imagery can be collected by imaging horizontally across the ground or from the air looking down. The airborne case generally has more turbulence near the object versus the camera and thus collects deep turbulence,[2] as shown in Fig. 1.

Modeling and simulating the effects of turbulence on imagery is typically done with wave propagation approaches, such as in Hardie et al.[3] Another approach is to use Zernike polynomials as basis functions[4] to model turbulence in the entrance pupil. Modeling approaches allow for a better understanding of how turbulence affects imagery and are an integral part of inverse modeling methods for mitigation. Additionally, they allow for truth frames to be available to quantifiably capture how well a mitigation algorithm performs when restoring the imagery.

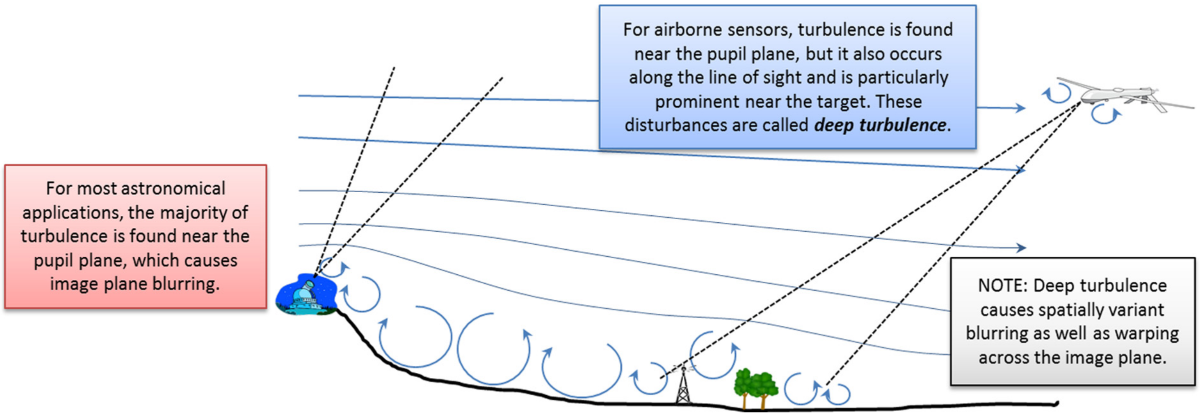
Turbulence mitigation approaches include, but are not limited to, lucky look,[5] speckle imaging,[6,7] and deblurring.[8] All three methods involve different processing steps to restore imagery; however, the use of PSF estimates in multiframe blind deconvolution (MFBD) approaches, such as in Ref. 8, is of particular interest for this paper. MFBD algorithms assume that the image frames have a common source, namely the object from which radiation propagated to the optical system, and they use this assumption to infer the turbulent PSFs impacting each image.[9] Each inferred PSF is then used to correct the corresponding distorted image frame. In this way, MFBD algorithms model the instantaneous atmospheric turbulence in the entrance pupil of the optical system. Thus, synthesis of a variety of PSFs not only helps model turbulence effects, but this process could also be used to provide blind deconvolution inputs into MFBD algorithms.

The rest of the paper is organized as follows: Sec. 1.1 provides background on the current wave optics approaches to simulate turbulence, and an outline of sparse and redundant signal representations in given in Sec. 1.2. Our dictionary representation of turbulence PSFs can be found in Sec. 2. Section 3 contains the statistical validation of the synthesized PSFs. Finally, our conclusions are discussed in Sec. 4.

1.1 *Modeling of Atmospheric Turbulence*

The current gold-standard method for modeling atmospheric turbulence effects is wavefront propagation. In general, the atmosphere is modeled by a series of phase screens with phase power spectral densities selected according to the atmospheric conditions being modeled, and then the

*Address all correspondence to: Bobby R. Hunt, E-mail: bhunt@integrity-apps.com; Amber L. Iler, E-mail: ailer@integrity-apps.com



For airborne sensors, turbulence is found near the pupil plane, but it also occurs along the line of sight and is particularly prominent near the target. These disturbances are called ***deep turbulence***.

For most astronomical applications, the majority of turbulence is found near the pupil plane, which causes image plane blurring.

NOTE: Deep turbulence causes spatially variant blurring as well as warping across the image plane.

Fig. 1 A comparison of turbulence cells, shown as arrows in this cartoon, and their proximity to both the focal plane and the object of interest, differs significantly for astronomical applications versus the "deep turbulence" case.

wavefront radiated from the object is propagated through these screens.[10,11] The physical impacts of the phase screens on the wavefront are calculated using geometrical optics, and the resulting phase distortions accumulated, along with any unperturbed phase that is propagating from the object.[12] These accumulated phase perturbations are combined to create a wavefront specific to that location in space, then the wavefront is similarly propagated onward through space to the next phase screen. Ultimately, all of these phase impacts are summed and appropriately scaled and then used to produce a simulated image impacted by turbulence.[12]

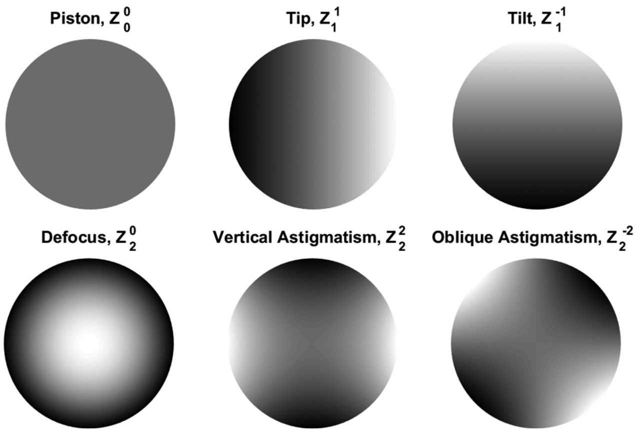
The summation of phase perturbations at the pupil plane of the optical system lends itself to representation by Zernike polynomials, which are defined in terms of radial and angular coordinates. Each of the Zernike terms describes an optical surface deviation related to classical aberration theory for optics. For example, the first six Zernike terms are shown in Fig. 2 along with their classical aberration names. Thus, by decomposing turbulent phase perturbations into Zernike polynomials, intuition can be gained into the dominating optical impacts of the atmosphere. However, due to the highly random nature of turbulence, turbulence is not a classical aberration, so the use of Zernike terms to characterize turbulent PSFs is not necessarily the optimal way to represent their impact on the object in the image plane, as has been seen in research for other ways of simulating turbulence.[13,14] For example, in deep turbulence cases with anisoplanatic effects across the field of view, it becomes necessary to include many more Zernike polynomial terms than those shown in Fig. 2, to characterize the turbulence, which reduces the computational compactness of the solution.

To optimize algorithmic approaches for modeling turbulence, we need to consider the desired objectives and the information available to the mitigation process. We know, for MFBD, image frames are gathered at the focal plane; therefore, we must have a detailed understanding of the focal plane itself. In addition, MFBD operates from a representation of the turbulent PSF structure as seen in the *focal plane*, since that is the only source of data available to the blind deconvolution process.

PSF synthesis in the focal plane is a distinction from the use of PSF synthesis in the pupil plane, as is done with wavefront propagation. For example, for Zernike-based phase screens derived from wavefront propagation, the phase modeling must be propagated to the optical pupil plane and then propagated to the image plane to form a PSF.[12] Ideally, we want to have efficient computation of PSFs that are capable of characterizing the full range of turbulent conditions directly in the image plane. Meeting this requirement will provide computational speed improvements in the forward modeling steps of mitigation methods, such as MFBD. These needs have led us to seek compact, easily computed methods of representing PSFs for long-range imaging simulations.

1.2 *Sparse and Redundant Representations in Signal Processing*

As effective as Fourier methods have been in signal processing, the past 20 years have seen rapid growth of alternative basis functions for signal processing, e.g., the wavelet transform.[15] Wavelet signal representations differ from Fourier representations by moving beyond the shift-invariant descriptions inherent in the Fourier model. Because wavelet models do not have a history from the eigen analysis of physical models, such as the wave-equation, wavelet representations of signals are sensitive to local structure in a signal. This local structure can be represented in terms of a hierarchy of signal structure describing how a signal has



Piston, Z_0^0

Tip, Z_1^1

Tilt, Z_1^{-1}

Defocus, Z_2^0

Vertical Astigmatism, Z_2^2

Oblique Astigmatism, Z_2^{-2}

Fig. 2 The first six Zernike polynomial terms characterize well-known classical optical aberrations.

significant components at different physical scales within the signal data being represented. The result is wavelet signal representation that is able to detect and capture how a signal structure changes with different hierarchies of signal measurement scales.

The latest departure from the space-invariant history of Fourier representations is the innovation usually referred to as compressive or compressed sensing.[16] Wavelet signal representations discarded the shift-invariant property of signal models, and compressed sensing models abandon use of the Shannon theorem (i.e., to not lose information for a uniformly sampled signal, we must sample at least two times faster than its bandwidth[16]) to constrain the number of samples necessary to faithfully represent a signal. The resulting paradigm of compressed sensing is to construct and use basis functions that are presumed to be more efficient in representing a signal, regardless of the Shannon criterion. Initial developments of compressed sensing demonstrated that a set of random vectors have favorable characteristics as basis functions for signal representation in the compressed paradigm.[17] A logical step, after this discovery, was to ask if a collection of basis functions could be constructed directly from the data available in a problem, so as to achieve an even more accurate representation of the data. This led to the development of methods for sparse and redundant representation of signals.[18]

The important properties of sparse and redundant representations are summarized in the K-SVD algorithm[19] used to construct a dictionary that represents the data. Sparse nomenclature convention refers to a single basis-like vector within a dictionary as an "atom." "Dictionary" is the sparse literature nomenclature for a collection of vectors, as a matrix, that can be used to represent any arbitrary signal vector. Thus, a sparse and redundant representation of any specific signal vector consists of the product of a coefficient vector with the dictionary matrix as

$$\mathbf{f} = D\mathbf{c}, \tag{1}$$

where D is the dictionary matrix, $\mathbf{c}$ is the coefficient vector, and $\mathbf{f}$ is the signal vector constructed by the inner product of D and $\mathbf{c}$.

The "redundant" appellation of the representation is related to the fact that the matrix D in Eq. (1) is not a collection of vectors that are orthonormal. Indeed, D not even needs to be square. For this reason, the representation of Eq. (1) is not unique for any specific vector $\mathbf{f}$. There can be multiple representations of a single vector, thus, "redundant" representations. Further, the entries of the coefficient vector $\mathbf{c}$, multiplied into the dictionary matrix D, are called "sparse," indicating that most of the entries of the coefficient vector $\mathbf{c}$ are zero, or so small as to be negligible in practice. Because the coefficient vectors are sparse, the representation of a vector $\mathbf{f}$ in this form is efficient, i.e., substantial compression of data is achieved when a vector is known in its dictionary form. This is also the origin of the term compressive sensing.

The generality of sparse and redundant representations, due to relaxing the orthonormal characteristic of the dictionary atoms, means that a dictionary is not specified *a priori*. A question then immediately arises: how do we construct a dictionary, and can a dictionary be created that is optimal for a given type of data? This is the nature of the K-SVD algorithm. A set of data is accumulated and analyzed in a manner that combines aspects of data compression by the k-means algorithm for vector quantization (VQ),[20] and refinement of quantized vector selections by means of the singular value decomposition.[21] The result is an algorithm that starts with a set of training signals and achieves the best representation for each member of the training set with vectors that are as sparse as possible.

The goal of the research we report herein is directed to the problem of deep turbulence. What aspect of deep turbulence is the motivation for the application of sparse and redundant representations? The motivation for that representation is the nature of atmospheric turbulence. Atmospheric deep turbulence is distributed over an extensive volume of space, the region where radiation reflected or emitted from an object propagates to the entrance pupil of an optical system. To accurately model those effects using wavefront propagation requires many assumptions, e.g., assumptions concerning the number of distinct regions present to create wavefront disturbances and the characteristics of those regions in causing wavefront deviations. Since this is impossible, it is conventional to simplify by adopting an arbitrarily fixed set of distinct regions, and then seek to characterize the effects of those regions when accumulated in the optical entrance pupil, i.e., Zernike polynomials. Note that no direct observation of the effects of deep turbulence occurs, only an assumption that the accumulated Zernike representation is appropriate.

In contrast, we consider the effects of deep turbulence observed in the focal plane of an optical system. A point source observed through deep turbulence creates a PSF. If many such PSFs are recorded, then there is a volume of data that represents the conditions of the deep turbulence in a manner that is directly related to the formation of images observed through the turbulence. Further, a sparse representation of those PSFs will be efficient from the viewpoint of characterizing the available information, because the process of defining sparse representations inherently identifies the most efficient bases for describing the PSFs. Utilizing that data, and the direct linkage to observable effects, is the motivation for the research that we report herein.

2 Dictionary Representation of Turbulent Point Spread Functions

2.1 *"ISO" Dataset Collections*

In September of 2015, the EO Target Detection and Surveillance Branch of the Air Force Research Laboratory (AFRL/RYMT) conducted a series of experiments at the Wright Patterson Air Force Base (WPAFB) in Dayton, Ohio. These experiments included, among other goals, collecting measurements of the isoplanatic angle of point sources imaged over a 5-km path. For this reason, we refer herein to this data as the "ISO" dataset. Two point sources were positioned on the ground and were imaged from the lowest floor of a tower, making the path nearly horizontal, and subject to substantial turbulence upwelling directly from the ground underneath the path of radiation propagating from the point sources. A perspective image of the tower from the target site is shown in Fig. 3(a). Collections occurred at different times of day, which resulted in significant changes in the turbulence conditions as solar heating changed the upwelling

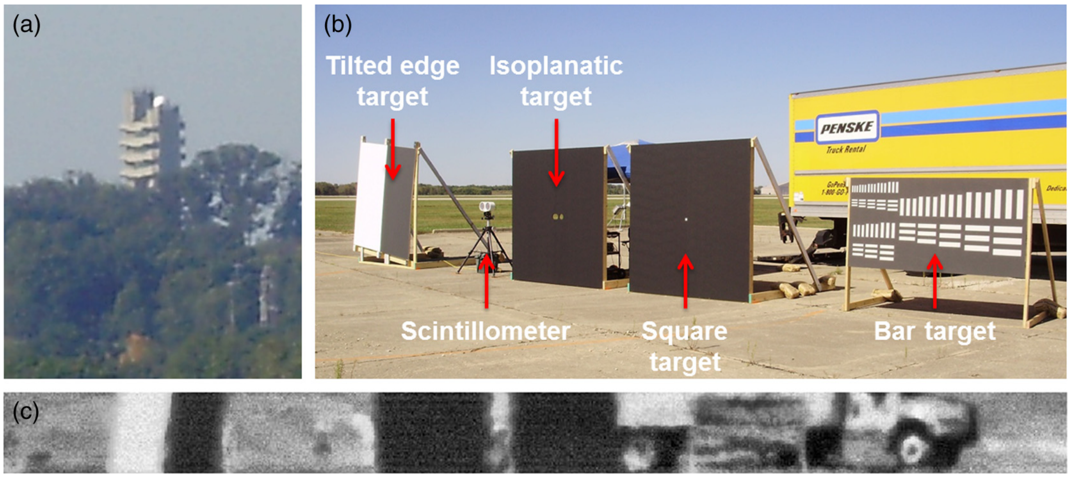


Fig. 3 (a) An image of the WPAFB tower from the ISO target site. (b) Configuration of the target array located ~5 km from the tower. (c) A sample image frame with left and right edges cropped to highlight the target array.

turbulence, and uncontrollable diurnal weather caused alterations in atmospheric variables such as wind, temperature, and humidity.

The ISO point sources were produced by two laser diodes, emitting at 785-nm wavelength, and projected through a collimator toward the tower. One of the laser sources was capable of being displaced with respect to the other, thus providing point sources at different angular positions with respect to the imaging receiver in the tower. This is why the isoplanatic target in Fig. 3(b) appears to have two holes parallel to the ground. The receiver in the tower was a simple refractor telescope with a primary entrance pupil diameter of 132 mm and a focal length of 925 mm. Aperture stops were used to produce entrance pupils of 38.1, 57.15, 63.5, 69.85, and 76.2 mm, respectively. The images formed by the telescope were collected by a high framerate (>1 kHz) PCO-Tech pco.edge 5.5 sCMOS scientific quality camera, with a focal plane of 2560×2160 pixels. An example image frame is shown in Fig. 3(c). The focal plane was windowed down to 51 rows by 1920 columns to collect at peak framerates. The pixels of the camera were square, 6.5 microns on a side. The pulse rate of the laser diodes and the frame rate of the camera were set to capture several hundred laser pulses within each data collection of 3000 frames. This created large data sets with many observations of the point sources, which imaged in the focal plane as PSFs of the turbulent characteristics.

A total of 14 ISO collections occurred on September 21 to 22, 2015: one collection on September 21, 2015, and 13 collections on September 22, 2015, each with 3000 frames of imagery. The collection on September 21 took place at 13:53 local time. The collections on September 22 were spaced from midmorning (09:45) to midafternoon (17:03). A scintillometer was present in the same test space, as shown in Fig. 3(b), and was used to measure data for the computation of C_n^2 and the values of the fried parameter, r_0, for all the data collections. The computations of C_n^2 and r_0 values were done with spherical propagation to the receiver telescope. The 14 collections of point source images possessed a range of r_0 values from 0.02 to 0.04 m. After inspection of the point source imagery for representativeness, quality and noise, and review of various statistics related to the collection times and conditions, the decision was made to select the collection dataset 12 from the collections occurring on September 22. This dataset became the source of PSF images to use for the construction of a sparse and redundant representation dictionary for turbulent PSFs. The reason for the selection of this set was the combination of the large aperture stop used on this collection (the 76.2-mm aperture), above average signal-to-noise ratio, and above average value of r_0 (0.031 m).

2.2 *Construction of a Sparse Dictionary from Point Spread Function Data*

All of the image frames collected during ISO tests were 51 rows by 1920 columns. The region in each of the frames of dataset 12 that contained laser point sources was identified and the PSFs of the point sources were extracted as 32×32 pixel subimages. A total of 200 of these 32×32 images of PSFs were randomly selected from the ISO dataset 12. Of the 200 PSF images, 100 were chosen to train a dictionary for PSF representation. The remaining 100 PSF images were reserved for testing PSFs constructed from the dictionary, as we describe in detail below.

The PSF images possessed visual noise, related to detector read-out and the number of electron counts actually generated in each detector pixel. A noise cleaning step was applied, based on measurements of the background noise level in areas near the edges of the 32×32 pixel PSF images. A threshold procedure was then used to set noise below a chosen power level to zero and normalize the PSFs to have unity total volume.

The 100 PSF images chosen for training a dictionary were processed by a MATLAB version of the K-SVD algorithm. The actual algorithm MATLAB code was obtained from a software package that accompanies the sparse representation textbook written by Elad.[20] The K-SVD algorithm is best understood by reference to the k-means algorithm, a well-known technique for training a codebook for data compression by VQ. A VQ codebook contains a set of vectors and is used to compress data by finding the single entry in the codebook that is closest, in the sense of nearest neighbor by Euclidean metric, to a sample of data that is to be

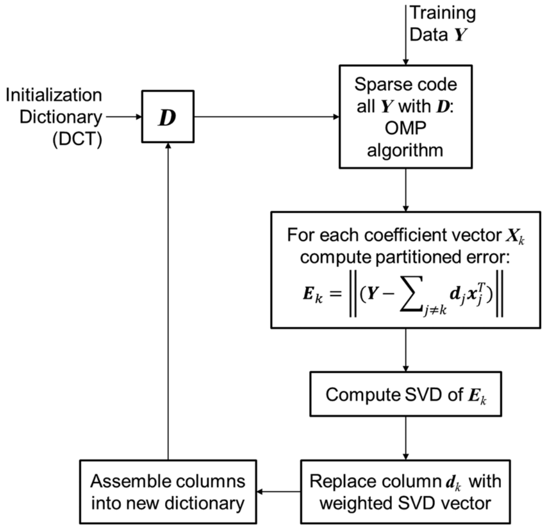

Training Data Y
Initialization Dictionary (DCT)
D
Sparse code all Y with D: OMP algorithm
For each coefficient vector X_k compute partitioned error:
$E_k = \left\|(Y - \sum_{j\neq k} d_j x_j^T)\right\|$
Compute SVD of E_k
Replace column d_k with weighted SVD vector
Assemble columns into new dictionary

Fig. 4 The K-SVD algorithm for dictionary construction uses singular value decomposition (SVD) to refine the dictionary elements based on representative training data.

compressed. The K-SVD algorithm is a generalization of the basic k-means algorithm because it constructs data as a linear combination of a small number of elements, rather than one element.

Figure 4 is a block diagram of the K-SVD algorithm. We summarize the basic steps of the algorithm in the following, and refer the reader to the original paper[19] for the complete details.

- *Choose an initial dictionary,* **D** In our case, the discrete cosine transform (DCT) algorithm was used to initialize a dictionary matrix, with all columns normalized to unity, for the training of the PSF dictionary. The size of the initial DCT dictionary must match the corresponding dimensions for the PSF dictionary that will be constructed during training. For example, if 256 dictionary elements are to result from training, then the DCT dictionary data must have 256 elements, corresponding to 16 different frequency resolutions in each of the two dimensions (horizontal and vertical) of the PSFs being represented in the dictionary.
- *Sparse coding step* For each training set vector, **y**, the representation error of that vector in the dictionary is defined over a set of coefficients, **x**, on the dictionary

$$e_i = \|\mathbf{y}_i - D\mathbf{x}_i\|_2^2. \tag{2}$$

  This error is minimized using a "greedy" pursuit algorithm, known as orthogonal matching pursuit (OMP),[22] subject to the requirement that each coefficient vector have no more than a chosen number of nonzero entries. In training the PSF dictionary, we experimented with different values for the number of nonzero dictionary elements and found that 256 entries captured a suitable degree of PSF structure that led to accurate PSF synthesis when applied to PSF representations. Each dictionary element was a matrix of size 32 × 32, which is 1024 values in a vector form, and matches the size of PSFs in the training set. Since 256 dictionary elements were trained, the final dictionary was of a matrix of size 1024 × 256.
- *Dictionary update step* The columns, d_k, of the dictionary, D, are selected, $k = 1,2,\ldots,256$. The overall representation error for each column, k, over the entire set of training data, **Y**, is defined as

$$\left\|(\mathbf{Y} - \sum_{j\neq k} d_j\mathbf{x}_j^T) - d_k\mathbf{x}_k^T\right\|, \tag{3}$$

  where superscript T denotes transpose from column to row. Here, we see that the total representation has been broken into two components, one with all the dictionary columns except the k'th, and the k'th itself. The first two terms, of **Y** and the sum of all errors except for the k'th column, are defined as the error E_k as shown in Fig. 3, which is due to all the columns of D except the k'th column. Further, the summation is a set of rank-one matrices. The update step computes the SVD of E_k. The first column of the SVD decomposition is weighted by the SVD coefficients and retained as the update to the column k.
- *Repeat sparse coding and dictionary update* The process returns to step 2, the sparse coding step, and continues for a preselected number of cycles or until the changes fall below a selected minimum.

This algorithm is intensive in computation time, as much as an hour on a typically available laptop workstation; however, it only needs to be done once to construct the desired dictionary. After construction of the dictionary, it is not needed in representing PSFs for as many times as desired.

Figure 5 is the result of using 100 ISO PSFs and the K-SVD algorithm to compile a dictionary of sparse and redundant basis functions. The dictionary elements are all 32 × 32 pixels in dimension, which corresponds to the extracted size of 32 × 32 PSFs from the ISO data. Note that the dictionary elements have properties that resemble PSFs, i.e., they are localized in space and have variations of structure within the localized space of an element. However, these are not PSFs but consistent structure in the training set PSFs. Bright white values are positive, dark black values are negative, and middle gray tones are near zero. Through selection of a linear combination of these elements, any of the PSFs in the training set can be represented. Further, linear combinations of elements such as these can be used to synthesize a PSF that is not in the training set nor any other set of PSF data. Using this dictionary for PSF synthesis is the benefit we achieve from the dictionary analysis and training procedure.

2.3 *Synthesis of Point Spread Functions from a Dictionary*

In this paper, the dictionary compiled from the training set PSFs will be referred to as a "custom" PSF dictionary, indicating that it is not a dictionary that would be suited for more general purposes. The DCT dictionary, used to initialize the K-SVD algorithm, is a general dictionary, and the generality of that dictionary is demonstrated by the presence of DCT

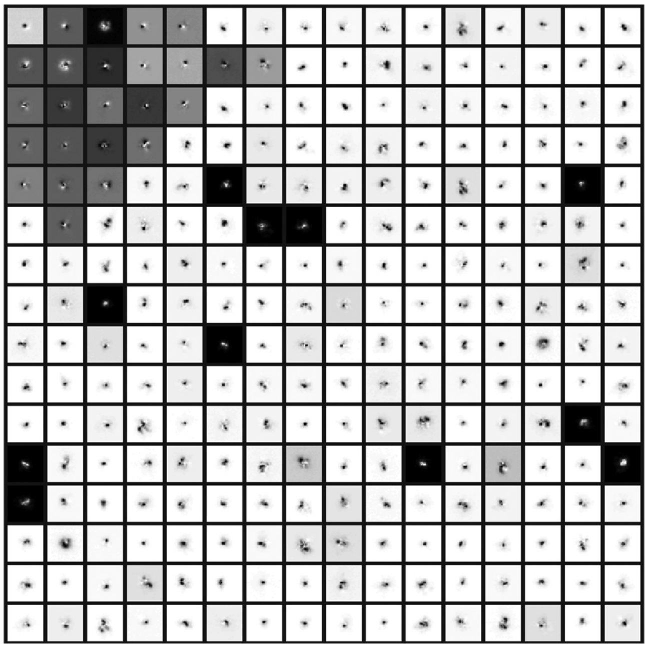


Fig. 5 The 256-element custom dictionary shown in this display was trained from the WPAFB ISO PSF data and highlights properties that are similar to PSFs.

representations in the standards of the JPEG image compression algorithm. The custom PSF dictionary, as shown in Fig. 5, has lost that generality. The loss of generality in the custom PSF dictionary is offset with a different benefit, however. Because the dictionary of Fig. 5 was created by analysis of PSFs collected in observations of turbulent atmosphere conditions, it is an obvious hypothesis that this dictionary can be used to synthesize PSFs that possess the characteristics of the turbulence that was used in the K-SVD algorithm.

The nature of a dictionary to represent data is seen in Eq. (1) in Sec. 1.2; namely, a coefficient vector multiplies a dictionary to construct a function. Therefore, it is a reasonable hypothesis that multiplication of the custom PSF dictionary by a coefficient vector will generate a function that possesses PSF properties. But the selection of the elements of the coefficient vector must be constrained. For example, multiplying a DCT dictionary be a set of random coefficients will almost never generate a meaningful or recognizable image, even though the DCT representation of images is well established by the JPEG algorithm. Thus, we must have a rationale by which to construct coefficients that will create a function with the characteristics of a PSF.

As an initial test, we used the custom PSF dictionary of Fig. 5 to represent all the PSFs in the training set. This was achieved using the OMP algorithm.[22] The OMP algorithm was set to make PSF representations using 64 dictionary elements (or "atoms," to use the nomenclature of sparse representations). We found that the 256-atom dictionary was so efficient at capturing PSF structure that a 64-coefficient representation was acceptably accurate in representation of an arbitrary PSF. Thus, OMP representation of the training set generated 100 sets of 64 coefficient values from each of the 256 dictionary atoms.

However, because of the efficiency of the representation with 64 atoms, not all of the coefficient atoms were used with equal frequency. The setting of 64 atoms in the OMP representation results in OMP choosing those 64 dictionary elements that are best suited for PSF representation by the OMP algorithm for each training set PSF, not distributing the usage of 64 atoms in a predetermined way over the entire 256 atoms of the full dictionary. We can consider the frequency in which atoms are actually used as characteristic of how training set PSFs are utilized in a specific set of PSF data. From this, it is possible to compile statistics of the dictionary coefficients for each dictionary atom. The statistics consist of the mean and standard deviation of the coefficients used in the representation of the training set PSFs.

Next, we use the statistics to construct PSFs, as shown in Fig. 6. We draw a PSF at random from the training set and refer to it as the "BasePSF." For the BasePSF, we know the 64 coefficients used to represent that PSF by OMP. We also know statistics of the 64 coefficients, compiled from all PSFs in the training set, which were used in that PSF. To synthesize a PSF, we know that we must perturb the coefficients of the BasePSF, but it is not acceptable to perturb the BasePSF coefficients in an arbitrary way, for the same reason that a random weighting of atoms in a DCT dictionary will not give a meaningful image. Thus, we perturb the coefficients by adding a random variable to the BasePSF coefficients, which is consistent with the mean and standard deviation for coefficients associated with the dictionary atoms. We complete the synthesis by multiplying the perturbed coefficient vector into the custom PSF dictionary, i.e., as in Eq. (1) above. Figure 6 provides a summary of the complete process.

3 Testing of Synthesized PSFs for Statistical Properties Consistent with ISO Source PSFs

The procedure described above produces PSFs derived from actual PSF data. This is different than methods of PSF synthesis that have been used in the past. Previous synthesis methods of PSFs for turbulent atmospheric conditions use a C_n^2 profile and wave propagation software to propagate a random wavefront perturbation of Zernike polynomial coefficients.[3,12,14] This is computationally intense. The synthesis procedure presented in Fig. 6 has the virtue of being extremely rapid, once the basic computations of the K-SVD algorithm have created a custom PSF dictionary. Once the first three steps in Fig. 6 are completed; the results can be reused in all future PSF syntheses with no further

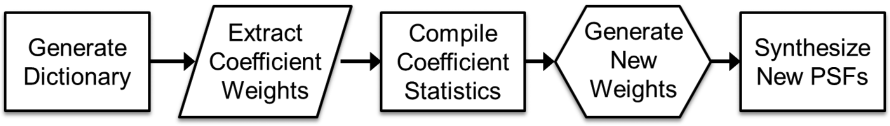


Fig. 6 To synthesize PSFs from the custom PSF dictionary, statistical characteristics of the training coefficients must be maintained.

computation cost. Therefore, we focus on the computation cost of the last two steps. The basic computation requirement for synthesis is efficient: draw random numbers and compute the matrix vector product of Eq. (1). In comparisons of a Zernike-based PSF wave propagation synthesis code to the custom PSF dictionary synthesis described above, the dictionary synthesis was at least three orders of magnitude faster in computation than the propagation method.

This comparison of execution times for Zernike and dictionary-based PSF synthesis has to be understood within the usual caution for direct comparisons of computed results. The same algorithm can be implemented in different high-level language software, with different computational architecture paradigms, in different operating systems on different computers, leading to significant differences in the execution time. In the case of Zernike polynomial models of turbulence, the major computational cost to simulate a turbulent wavefront at the entrance pupil is to migrate a wavefront that is initially uniform in phase through multiple phase screens that randomly distort the phase, each phase screen being a distance from its neighbors. The wavefront is migrated by wave-optics propagation, e.g., Fresnel, and requires many evaluations of the Fresnel kernels over discrete steps of distance. Thus, the total number of propagation steps, and the number of samples across the wavefront, determines the computational effort and can be larger or smaller according to parameter choices.

The above experience, of three orders of magnitude of difference in our synthesis compared with wavefront propagation, was accumulated on the same computer, running MATLAB for both Zernike and dictionary synthesis. The number of samples of the synthesized PSFs was the same in both cases, and the number of phase screens and the total number of propagation steps were based on previous experience in simulating turbulence for development of MFBD algorithms. The Zernike-based code has been in use for several years in MFBD processing algorithms and is considered reliable. But neither the Zernike nor dictionary code was extensively profiled and optimized. For these reasons, we have confined our comparative disclosure of performance to the order-of-magnitude estimates seen immediately above and elsewhere in this paper.

No matter how fast, the dictionary synthesis method is of no utility if it produces PSFs that are not realistic or do not possess valid PSF characteristics. Thus, we carried out a series of tests to determine how PSFs synthesized by the dictionary method compare in relation to the real PSFs that were used to compile the dictionary. As stated above in Sec. 2.2, a total of 200 PSFs were extracted from the ISO collection dataset 12 PSF images, with 100 used for the training set data that produced the dictionary data. The remaining 100 images were used as a source of actual PSF data to compare with the synthesized PSFs. We shall refer to the reserved data as the test set PSFs. The purpose of these comparisons was to determine if the synthesized PSFs possessed characteristics that were comparable with PSFs from the same atmospheric conditions, but which were never used in training the custom PSF dictionary. This was adopted as a means of verification for our synthesis procedure with independent data that could not have any connection to the dictionary that synthesized the PSFs. In the following, we describe the comparison tests we applied.

3.1 *Visual Inspection*

As described in Sec. 2.3, the dictionary synthesis method synthesizes PSFs through statistically based perturbation of the coefficients of a BasePSF. The first test, therefore, was to visually examine the synthesized PSFs to determine if they had excessive visual resemblance to the original BasePSF. The visual inspections were conclusive; the synthesized PSFs did not show any physical resemblance to the BasePSF. In other words, even though the synthesized PSFs were constructed from the same BasePSF, the perturbation process was sufficient to destroy any heritage of the BasePSF in the synthesized PSFs.

Further visual inspections were carried out to determine if the synthesized PSFs had any visible differences in overall structure and morphology from the actual test set PSFs. Figure 7 is an example of the results of such inspections. Two of the PSFs in Fig. 7 were synthesized from the algorithm, and two of the PSFs were drawn at random from the 100 PSFs that were not used to construct the custom PSF dictionary. The reader is invited, for their own amusement, to make a guess at which PSFs are synthetic and which are real. The answer is in the Appendix.

3.2 *Ensemble Average of Point Spread Functions*

Following the visual inspection, the next simplest test was to compute the ensemble average of 100 synthesized PSFs and the ensemble average of the 100 test set PSFs reserved for comparison purposes. The ensemble average is simple: all PSFs, synthesized and from test set collections, are 32 × 32 pixel images. All 100 synthesized PSFs were averaged at each of the 32 × 32 pixel grid positions, and likewise for the 100 test set PSFs not used for training the custom dictionary. Figure 8 shows a side-by-side comparison of the ensemble averages for the synthesized and test set PSFs. The two ensemble images appear extremely similar.

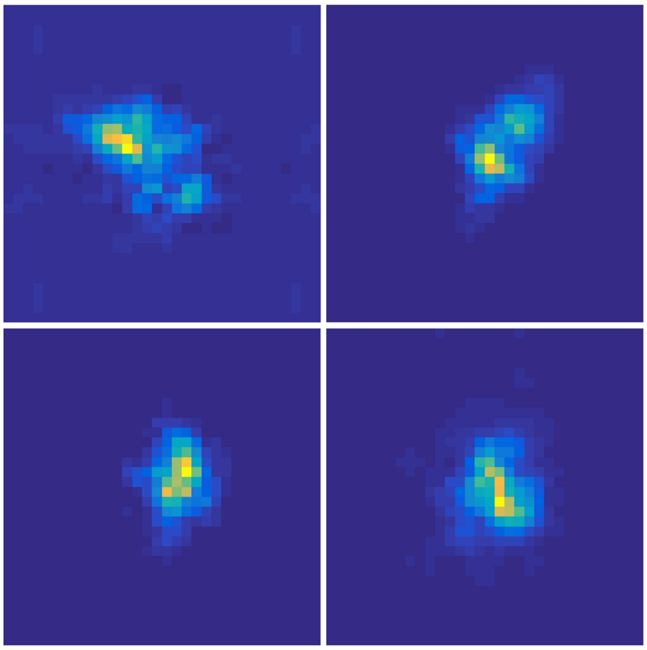


Fig. 7 Two synthesized and two test set PSFs. The reader is challenged to guess which are measured PSFs and which are synthesized. The answer can be found in the Appendix.

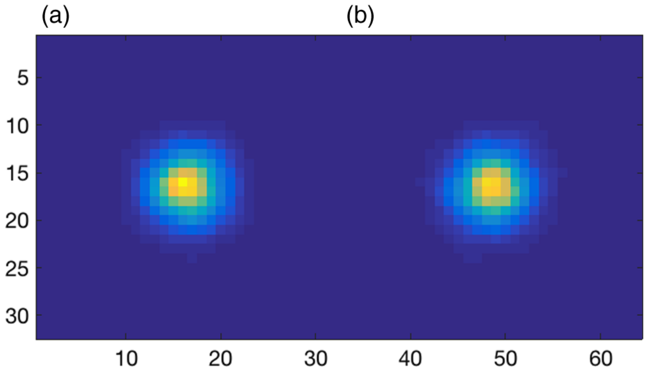


Fig. 8 (a) The similarity of the ensemble average for the test set PSFs and (b) synthesized PSFs supports the hypothesis that the synthesized PSFs, on average, mimic the turbulence conditions of the collected data.

A numerical comparison was also carried out, subtracting the synthesized ensemble average from the test set ensemble average. The result of the subtraction showed mean-square numerical differences <1 part in 10^3, which effectively corresponds to a signal-to-noise-ratio of ~26 dB of the synthesized PSFs compared with the test set PSFs.

3.3 *Average Autocorrelation of Point Spread Functions*

The second averaging test was to calculate the average autocorrelation of the synthesized and test set PSFs. For the synthesized and test set PSFs, each 32×32 pixel PSF image was embedded in a 64×64 frame of zeros, and then autocorrelated by FFT calculation. The 100 autocorrelations of the synthesized and test set PSFs were then averaged and the averages compared. Figure 9 is a display of the average autocorrelations. The mean-square difference between the average autocorrelations of synthesized and test set PSFs was ~1 part in 10^4, which is effectively a signal-to-noise-ratio of ~40 dB of the synthesized PSF autocorrelations compared with the original PSF autocorrelations.

3.4 *Distribution of Point Spread Function Intensities*

Since synthesized PSFs have similar structure, when compared visibly, with actual ISO test set PSFs, the motivation arises to do more quantitative analysis to determine if the visible similarities are supported by statistical tests. The first such test was the distribution of intensity in the test set PSFs and the synthesized PSFs. The two-sided Kolmogorov–Smirnov (KS2) test, which is based on the empirical probability density functions of two different random variables as estimated by histograms, was used to test the distribution of intensities, using a significance level of 0.01. The test declared that the synthesized PSFs had the same statistical distribution of intensity.

3.5 *Distribution of Central Moments*

The next statistical test was the distribution of PSF intensities about the centroid of the PSFs. This was performed by computing the centroid of synthesized and test set PSFs and then using the centroid for each PSF to compute the central moment about the centroid, thus measuring the dispersion of each synthesized and test set PSF. The results were then tested by the KS2 test to determine if the central moment statistics were the same. The KS2 test declared the two sets of central moments came from the same distribution at the 0.01 level.

3.6 *Distribution of Strehl Ratio*

A final test was the computation of the Strehl ratios for the two sets of PSFs. The Strehl ratio, as defined in terms of

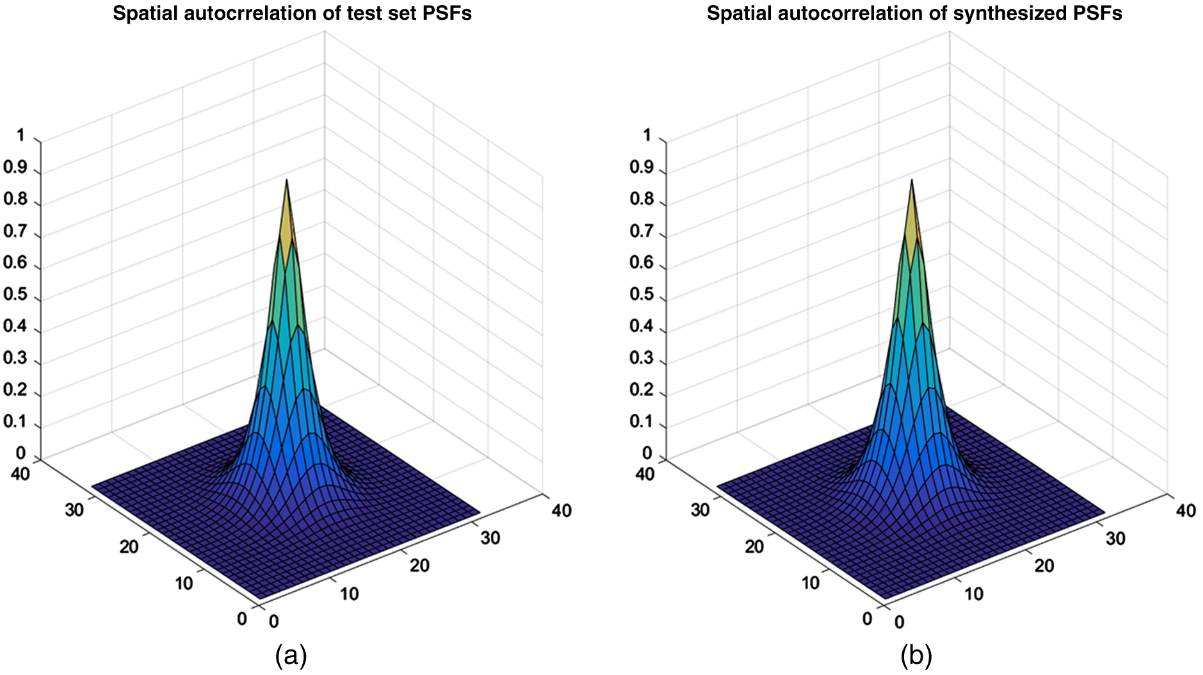


Fig. 9 (a) The average autocorrelations for test set PSFs and (b) synthesized PSFs are visually indistinguishable.

spatial frequency, was used.[23] Again, the KS2 test was applied to the empirical probability distribution of Strehl ratio values and confirmed the two distributions were the same at the 0.01 significance level.

4 Conclusions

In conclusion, we have shown that sparse and redundant representations of turbulent PSFs is a highly efficient way of characterizing PSFs in the focal plane. Using point source data collected under deep turbulence conditions, we constructed a custom sparse dictionary containing 256 atoms and verified its ability to represent turbulent PSF data using 64 coefficients or less. Analysis of the 64 coefficients confirmed it was reasonable to multiply the custom PSF dictionary by a coefficient vector to synthesize imagery that possesses PSF properties. This mathematically simple operation bypasses the need for Fourier analysis to synthesize atmospheric turbulence conditions.

However, generation of the coefficient vector could not be done in a random way. The mean and standard deviation of the 64 coefficients used to represent each of the training set PSFs were derived and became guidelines for generating weighting coefficients. Thus, to synthesize a PSF, we perturbed the coefficients of a BasePSF by adding random variables, which were consistent with the same mean and standard deviation of coefficients associated with the dictionary atoms, to the BasePSF coefficients. Synthesis is completed by multiplying the perturbed coefficient vector into the derived custom PSF dictionary.

We verified PSFs synthesized using our dictionary synthesis process were statistically representative of the turbulence conditions used to construct the dictionary through several tests:

1. Simple visual inspection confirmed similar synthesized PSF characteristics to other data collected under similar conditions without duplication of physical features.
2. The ensemble average PSF for synthesized PSFs was numerically equivalent to the ensemble average for the test set PSFs to the 26-dB level.
3. The average autocorrelations of synthesized and test set PSFs were equivalent within a signal-to-noise-ratio of ∼40 dB.
4. A KS2 test of synthesized and test set PSFs confirmed the same statistical distribution of intensity for the two PSF ensembles.
5. A KS2 test of the distribution of the PSF central moment declared the synthesized and test set central moments came from the same distribution.
6. The empirical probability distribution of Strehl ratio values for the synthesized and test set PSFs verified the two distributions were the same at a KS2 0.01 significance level.

Finally, with the success shown herein and because PSF calculation directly in the focal plane removes the need for Zernike analysis in the pupil plane, we believe these methods will prove to be a valuable addition to turbulence modeling methods. Initial results using dictionary synthesis showed three orders of magnitude computational improvement for synthesizing turbulent atmosphere conditions over conventional optical wave propagation simulation. This comes as no surprise, since sparse and redundant representations characterize volume turbulence in a manner directly related to the formation of images observed through atmospheric turbulence.

Utilization of this approach, with its direct linkage to observable atmospheric turbulence effects, will motivate future research into these methods. In particular, we acknowledge that the PSF synthesis technique discussed in this paper focuses on generating synthetic PSFs for a specific real-world dataset. Thus, in the spirit of this paper, real-world datasets are explicitly needed to build different dictionaries. However, additional work to expand this technique beyond the limiting conditions of a specific real-world dataset is underway and results at hand suggest it may be possible to scale synthesized PSFs in a way that would provide more varied conditions of turbulence, including variations in wavelength of illumination or observation. These results, when completed, will be disclosed in a future paper extending the initial results reported herein.

Appendix: Visual PSF Comparison

In Fig. 7, the upper left and lower right PSFs were synthesized from the custom PSF dictionary by the algorithm described in Sec. 2.3.

Acknowledgments

This work was funded by the Air Force Research Laboratory, Sensors Directorate, under contract FA8650-12-D-1344. The authors would like to thank the AFRL EO Target Detection and Surveillance Branch (AFRL/RYMT) for the opportunity to pursue this research and for their invaluable contributions and ideas during the planning phases of this effort. This paper has been approved for public release with the PA Approval #88ABW-2017-3952.

References

1. M. C. Roggemann and B. M. Welsh, *Imaging Through Turbulence*, CRC Press, Boca Raton, Florida (1996).
2. R. G. Paxman et al., "Spatial stabilization of deep-turbulence-induced anisoplanatic blur," *Opt. Express* **24**(25), 29109–29125 (2016).
3. 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).
4. F. Sroubek and J. Flusser, "Multichannel blind iterative image restoration," *IEEE Trans. Image Process.* **12**(9), 1094–1106 (2003).
5. M. Aubailly et al., "Automated video enhancement from a stream of atmospherically-distorted images: the lucky-region fusion approach," *Proc. SPIE* **7463**, 74630C (2009).
6. T. W. Lawrence et al., "Extended-image reconstruction through horizontal path turbulence using bispectral speckle interferometry," *Opt. Eng.* **31**(3), 627–636 (1992).
7. C. J. Carrano, "Speckle imaging over horizontal paths," *Proc. SPIE* **4825**, 65–75 (2002).
8. R. C. Hardie et al., "Block matching and Wiener filtering approach to optical turbulence mitigation and its," *Opt. Eng.* **56**(7), 071503 (2017).
9. C. L. Matson et al., "Fast and optimal multiframe blind deconvolution algorithm for high-resolution ground-based imaging of space objects," *Appl. Opt.* **48**, A75–A92 (2009).
10. J. M. Martin and S. M. Flatté, "Intensity images and statistics from numerical simulation of wave propagation in 3-D random media," *Appl. Opt.* **27**, 2111–2126 (1988).
11. J. Schmidt, *Numerical Simulation of Optical Wave Propagation with Examples in MATLAB*, SPIE Press, Bellingham, Washington (2010).
12. J. P. Bos and M. C. Roggemann, "Technique for simulating anisoplanatic image formation over long horizontal paths," *Opt. Eng.* **51**(10), 101704 (2012).

13. E. Repsai and R. Weiss, "Analysis of image distortions by atmospheric turbulence and computer simulation of turbulence effects," *Proc. SPIE* **6941**, 69410S (2008).
14. S. Lachinova et al., "Comparative analysis of numerical simulation techniques for incoherent imaging of extended objects through atmospheric turbulence," *Opt. Eng.* **56**(7), 071509 (2017).
15. S. Mallat and A Wavelet, *Tour of Signal Processing*, 2nd ed., Academic Press, San Diego, California (1999).
16. R. Baraniuk, "Compressive sensing (lecture notes)," *IEEE Signal Process. Mag.* **24**(4), 118–121 (2007).
17. D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory* **52**(4), 1289–1306 (2006).
18. M. Elad, *Sparse and Redundant Representations*, Springer-Verlag, New York (2010).
19. M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.* **54**(11), 4311–4322 (2006).
20. A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Press, Norwell, Massachusetts (1992).
21. C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs, New Jersey (1974).
22. J. Tropp, "Greed is good: algorithmic results for sparse approximation," *IEEE Trans. Inf. Theory* **50**(10), 2231–2242 (2004).
23. L. C. Roberts et al., "Is that really your Strehl ratio?" *Proc. SPIE* **5490**, 504–515 (2004).

Bobby R. Hunt is a senior technical advisor with Integrity Applications Incorporated. He received his BS degree in aeronautical engineering from Wichita State University in 1964, his MS degree in electrical engineering from Oklahoma State University in 1965, and his PhD in systems and electrical engineering from the University of Arizona in 1967. He is the author of 180 publications in scholarly archival journals, conference proceedings, and books. His current research interests are mitigation of atmospheric turbulence, shadow imaging of space objects, and identification and classification of unresolved space objects. He is a life fellow of IEEE and a fellow of the Optical Society of America.

Amber L. Iler is a principal scientist at Integrity Applications Incorporated. She received her BS degree in astronomy and music from the University of Michigan in 1994 and her MS degrees in physics and astronomy, respectively, from Eastern Michigan University in 1996 and the University of Massachusetts in 1999. She is the author of more than 15 scientific publications and is a US patent holder. Her current research interests focus on EOIR physics, with special emphasis on exploitation of remote sensing imagery data. She is a member of SPIE.

Christopher A. Bailey is an electro-optical engineer at Leidos in Dayton, Ohio. He received his BS degree in engineering physics and his PhD in chemical physics from the University of Central Oklahoma in 2003 and Kent State University in 2008, respectively. He is the author of more than 20 journal papers and has coauthored one book chapter. His current research interests include optical system optimization, image restoration in turbid media, and physics-based optical modeling of laser based and passive imaging systems.

Michael A. Rucci is a research engineer at the Air Force Research Laboratory, Wright-Patterson AFB, Ohio. He received his MS and BS degrees in electrical engineering from the University of Dayton in 2014 and 2012, respectively. His current research includes day/night passive imaging, turbulence modeling and simulation, and image processing.