Little publicly available data exists for polarimetric measurements. When designing task specific polarimetric systems, the statistical properties of the task specific data becomes important. Until better polarimetric datasets are available to deduce statistics from, the statistics must be simulated to test instrument performance. Most imaged scenes have been shown to follow a power law power spectral density distribution, for both natural and city scenes. Furthermore, imaged data appears to follow a power law power spectral distribution temporally. We are interested in generating image sets which change over time, and at the same time are correlated between different components (spectral or polarimetric). In this brief communication, we present a framework and provide code to generate such data.

2. THEORY

Statistical distributions [10–13] can be generated via a variety of methods. We use the convolution/Fourier transform method here [13, 14], whereby some white noise distribution is generated, then filtered via convolution or multiplication in the Fourier domain. This method is straightforward to implement and results in fast computational times. The noise can also be correlated, correlation of Gaussian white noise is straightforward to compute and generate [15], however this does not necessarily translate into the easily generated correlated noise with a specific covariance matrix and a specific distribution [12]. In this communication we present algorithms to generate samples of specific PSD distributions with a specific covariance matrix. Although the covariance matrix between polarimetric images is currently unknown, we can set reasonable values for polarimetric imaging system evaluation. Polarimetric scenes will have specific correlations between \( s_0, s_1, s_2, s_3 \), for different environments, however these correlations are still being determined. We intend to use large polarimetric datasets in the future to determine these correlations, and at that time we will refactor the sample generation code to represent a proper covariance matrix.

In this communication we present 3 algorithms; 1) generation of a number of images which obey a power law PSD spatially, but are temporally uncorrelated, 2) generation of a number of images which obey a power law PSD spatio-temporally, 3) generation of \( N \) sets of images which obey a power law PSD spatio-temporally and are correlated between the sets. The power law PSDs are generated in the following way:

- images or sets of images of white noise are generated.
- the images are are taken to the frequency domain using a 2-dimensional or 3-dimensional fast Fourier transform (FFT) depending on the desire for a temporal power law PSD.
- Fourier domain filters are generated by using the power law in Eq. (1.1) where \( f = \sqrt{\xi^2 + \eta^2} \) or \( f = \sqrt{\xi^2 + \eta^2 + \nu^2} \) and \( \xi \) corresponds to the spatial frequencies in the \( x \)-direction, \( \eta \) corresponds to the spatial frequencies in the \( y \)-direction, and \( \nu \) corresponds to the temporal frequencies in the \( z \)-direction.

\[
\frac{A}{|f|^\gamma} \quad (1.1)
\]
\[ x_{n_1,n_2,n_3} = \frac{1}{N_1 N_2 N_3} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} \sum_{k_3=0}^{N_3-1} e^{2\pi i (n_1 k_1 / N_1 + n_2 k_2 / N_2 + n_3 k_3 / N_3)} F(k_1,k_2,k_3) X_{k_1,k_2,k_3} \]  

(1.2)

frequencies.

**A. Mean and variance**

The Fourier transform of normally distributed Gaussian noise is again normally distributed Gaussian noise [16]. Once filtered by a power law PSD, the spatio-temporal data will have a specific mean and variance, which we would like to derive analytically and allow a user to specify. We present the derivation for white Gaussian noise (WGN) under a discrete Fourier transform (DFT) here. In the algorithms presented in this communication, we generate zero-mean WGN in the spatio-temporal domain, then transform to the Fourier domain for filtering. This results in zero-mean WGN in the discrete Fourier domain, with an associated Hermiticity condition. Each spatio-temporal frequency location can be treated as a random variable, \( X_{k_1,k_2,k_3} \) at location \((k_1,k_2,k_3)\) of the DFT domain. The inverse discrete Fourier transform is defined in Eq. (1.2) at the point \( x_{n_1,n_2,n_3} \). We can then compute the mean and variance of \( x_{n_1,n_2,n_3} \) assuming that the random variables \( X_{k_1,k_2,k_3} \) are independent with variance \( \sigma^2 \) and \( F(k_1,k_2,k_3) \) is the Fourier domain filtering function. If we assume that the mean of each \( X_{k_1,k_2,k_3} \) is 0, then the mean of \( x_{n_1,n_2,n_3} \) is also 0. The variance for \( x_{n_1,n_2,n_3} \) is then

\[ \text{Var}(x_{n_1,n_2,n_3}) = \frac{1}{(N_1 N_2 N_3)^2} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} \sum_{k_3=0}^{N_3-1} \frac{F^2(k_1,k_2,k_3) \sigma^2}{N_1 N_2 N_3} \].

(2.1)

Notice the lack of dependence on \((k_1,k_2,k_3)\) due to the white noise assumption. Additionally if zero mean WGN with a variance of 1 is used as the initial spatio-temporal input, then the DFT will have a variance of \( \sigma^2 = N_1 N_2 N_3 \) for the standard FFT definition. This allows us to analytically adjust the variance and mean of the generated datasets to specified values in our generation algorithms. Note that for a spatial distribution only Eq. (2.1) reduces to the 2-dimensional case.

**B. Correlation**

Stokes images can be thought of as a set of 4 spatio-temporal data cubes which are correlated. The third algorithm in the following section can generate \( K \) correlated spatio-temporal datacubes as samples from a specific 3-dimensional power law PSD. \( \Sigma \) is the input \( K \times K \) covariance matrix.

There is some subtlety involved in both specifying \( \Sigma \) and the mean and STD of each data cube. Given a vector \( \mathbf{X} \) of independent variables with 0 mean and variance 1, we have the covariance matrix given by

\[ \Sigma_{\mathbf{X}} = \mathbf{E} \left[ (\mathbf{X} - \mathbf{E}[\mathbf{X}]) (\mathbf{X} - \mathbf{E}[\mathbf{X}])^T \right] = \mathbf{E} \left[ \mathbf{XX}^T \right] = \mathbf{I} \]

where \( \mathbf{I} \) is the identity matrix. Then if \( \mathbf{Z} = \mathbf{LX} \) we have

\[ \Sigma_{\mathbf{Z}} = \mathbf{E} \left[ \mathbf{ZZ}^T \right] = \mathbf{E} \left[ \mathbf{LX} (\mathbf{LX})^T \right] = \mathbf{E} \left[ \mathbf{LXX}^T \mathbf{L}^T \right] = \mathbf{LL}^T, \]

(2.2)

where \( \mathbf{L} \) can be found from \( \Sigma \) via the Cholesky decomposition. In our application we specify \( \Sigma_{\mathbf{Z}} \), compute \( \mathbf{L} \), obtain a linear transformation of the datasets using \( \mathbf{Z} = \mathbf{LX} \), then add the specified means to obtain specific Stokes datasets.

**3. ALGORITHMS AND CODE**

We describe the specific algorithms and the associated Matlab code here. The first algorithm generates sets of random images with a spatial PSD given by

\[ \frac{A}{|f|^\gamma} \]

(3.1)

where \( f = \sqrt{\xi^2 + \eta^2} \). Note that in the algorithms below, we cannot evaluate the PSD filter for \( f = 0 \) since that results in an infinite value, so the \( f = 0 \) value is replaced with \( f = \varepsilon \) for some \( \varepsilon > 0 \). We can specify the number of images, \( N \), the desired mean parameter \( rB \), the desired standard deviation, \( rA \), the white noise generating function itself, NF CN, and the force positive parameter \( FP \). The \( FP \) parameter is for optical image sensors, which have non-negative measurements, if set to true, any negative values are truncated to 0. Note that if NF CN does not generate independent samples then the computed mean and variance may be incorrect.

The second algorithm generates a set of \( N \) images, but assumes that they obey a 3-dimensional power law PSD distribution [1] where \( f = \sqrt{\xi^2 + \eta^2 + \nu^2} \). This results in Brownian noise or derivatives of Brownian noise depending on the value of \( \gamma \). Statistics of natural images obey a spatial power law PSD where \( \gamma \approx 2 \) [3], with slightly different numbers for different types of images. This implies that we cannot use \( \gamma = N \) for our 3-dimensional filtering algorithm, where \( N \) is derived empirically for static images. The details about how the exponent in the
Algorithm 1. genObsPowerLaw. Note that the calculation for nVar does not square the denominator as shown in Eq. (2.1), this is due to the factor of numX · numY which is imposed on the variance from the 2-dimensional FFT.

1: functiongenObsPowerLaw(N,A,γ,rA,rB,NFCN,FP)
2: \( r_l \leftarrow NFCN(N) \)    \( r_l \) is a set of \( N \) random images with STD 1 and mean 0 when NFCN is WGN
3: \( F \leftarrow \sqrt[A]{f_{f_{1/2}}} \) \( \triangleright \) Take the square root to get the filter for a PSD sample
4: for \( k = 1 \) to \( N \) do
5: \( r_{IF}[k] \leftarrow FFT2(r_{I}[k]) \) \( \triangleright \) take 2-D FFT of the current image
6: \( r_{IF}[k] \leftarrow F \cdot r_{IF}[k] \) \( \triangleright \) Filter the image in the spatial frequency domain
7: \( r_{IF}[k] \leftarrow IFFT2(r_{IF}[k]) \) \( \triangleright \) take 2-D IFFT of the filtered image
8: \( nVar \leftarrow \sum(F \odot F)/(\text{numX} \cdot \text{numY}) \) \( \triangleright \odot \) here denotes elementwise multiplication. \( \sum \) sums over all elements.
9: \( r_{IF} \leftarrow (r_{A}/\sqrt{nVar}) \cdot r_{IF} + r_{B} \) \( \triangleright \) Specify the STD and mean of \( r_{IF} \), first normalizing back to an STD of 1 via \( nVar \).
10: if \( FP = \text{true} \) then \( r_{IF}[r_{IF} < 0] \leftarrow 0 \) \( \triangleright \) Any value less that zero is truncated to zero
11: return \( r_{IF} \) \( \triangleright \) Random image samples in the spatial domain with the specified PSD

Algorithm 2. genObsPowerLaw2. Note that the calculation for nVar does not square the denominator as shown in Eq. (2.1), this is due to the factor of numX · numY · N which is imposed on the variance from the 3-dimensional FFT.

1: functiongenObsPowerLaw2(N,A,γ,rA,rB,NFCN,FP)
2: \( r_l \leftarrow NFCN(N) \) \( r_l \) is a set of \( N \) random images with STD 1 and mean 0 when NFCN is WGN
3: \( F \leftarrow \sqrt[A]{f_{f_{1/2}}} \) \( \triangleright \) take 3-D FFT of the current image
4: \( r_{I_{FFT}} \leftarrow FFT3(r_{I}) \) \( \triangleright \) Filter the image in the spatio-temporal frequency domain
5: \( r_{I_{FFT}} \leftarrow F \cdot r_{I_{FFT}} \) \( \triangleright \) take 3-D FFT of the filtered image
6: \( nVar \leftarrow \sum(F \odot F)/(\text{numX} \cdot \text{numY} \cdot N) \) \( \triangleright \odot \) here denotes elementwise multiplication. \( \sum \) sums over all elements.
7: \( r_{IF} \leftarrow (r_{A}/\sqrt{nVar}) \cdot r_{IF} + r_{B} \) \( \triangleright \) Specify the STD and mean of \( r_{IF} \), first normalizing back to an STD of 1 via \( nVar \).
8: if \( FP = \text{true} \) then \( r_{IF}[r_{IF} < 0] \leftarrow 0 \) \( \triangleright \) Any value less that zero is truncated to zero
9: return \( r_{IF} \) \( \triangleright \) Random image samples in the spatio-temporal domain with the specified PSD

Algorithm 3. genObsPowerLaw3. Note that the matrix multiplication in line 12 treats the data as a large set of \( K \times 1 \) vectors and multiplies each vector by \( r_{A} \) can be eliminated because it is specified in the diagonal of \( \Sigma \).

1: functiongenObsPowerLaw3(N,A,γ,rB,NFCN,FP,K,Σ)
2: \( L \leftarrow \text{CHOL}(\Sigma) \) \( \triangleright \) Compute the Cholesky decomposition of the covariance matrix and take the upper triangular matrix
3: for \( k = 1 \) to \( K \) do
4: \( r_l[k] \leftarrow NFCN(N) \) \( \triangleright \) \( r_l \) is a set of \( K \) random image sets with STD 1 and mean 0 when NFCN is WGN
5: \( F \leftarrow \sqrt[A]{f_{f_{1/2}}} \) \( \triangleright \) Take the square root to get the filter for a PSD sample, \( f \) here is 3-D frequency.
6: \( nVar \leftarrow \sum(F \odot F)/(\text{numX} \cdot \text{numY} \cdot N) \) \( \triangleright \odot \) here denotes elementwise multiplication. \( \sum \) sums over all elements.
7: for \( k = 1 \) to \( K \) do
8: \( r_{I_{FFT}} \leftarrow FFT3(r_{I}[k]) \) \( \triangleright \) take 3-D FFT of the current image
9: \( r_{I_{FFT}} \leftarrow F \cdot r_{I_{FFT}} \) \( \triangleright \) Filter the image in the spatio-temporal frequency domain
10: \( r_{IF} \leftarrow IFFT3(r_{I_{FFT}}) \) \( \triangleright \) take 3-D IFFT of the filtered image
11: \( r_{IF}[k] \leftarrow (1/\sqrt{nVar}) \cdot r_{IF} \) \( \triangleright \) set the variance to 1.
12: \( r_l \leftarrow L \cdot r_{IF} \) \( \triangleright \) Matrix-vector multiply \( L \) by \( r_l \) as pages of \( K \times 1 \) vectors. The datacubes will be correlated as \( \Sigma \).
13: \( r_l \leftarrow r_l + r_{B} \) \( \triangleright \) Add the means. Added as a \( K \times 1 \) vector to the \( K \) pages.
14: if \( FP = \text{true} \) then
15: return \( r_{IF} \) \( \triangleright \) Random image samples in the spatio-temporal domain with the specified PSD
PSD changes for Brownian motion processes in different Euclidean dimensions is given by Bassingthwaighte and Raymond [17] and by Heneghan, Lowen, and Teich for the 2-dimensional case [18]. This communication will not delve into details, but the exponent is dependent on $H = E + 1/D$ where $E$ is the Euclidean dimension, $D$ is the fractal dimension of the process, and $H$ is the Hurst coefficient. This must be taken into account when using the 3-dimensional algorithm to maintain specific spatial PSD distributions. The PSD is proportional to

$$A \left| f \right|^{2H+1}. \tag{3.2}$$

4. STOKES DATA

In the previous section we presented 3 algorithms which provide samples of power law PSD distributions for a variety of situations. In this section we will use the GENOBSPowerLaw3 algorithm to generate samples as inputs for our analysis of Stokes parameter measuring instruments. We have selected specific $\Sigma, rB$ parameters for each Stokes parameter. The covariance matrix is

$$\Sigma = \begin{bmatrix} 500 & 200 & 200 & 100 \\ 200 & 350 & -100 & 50 \\ 200 & -100 & 350 & 50 \\ 100 & 50 & 50 & 200 \end{bmatrix} \tag{4.1}$$

and

$$rB = \begin{bmatrix} 200 \\ 0 \\ 0 \\ 0 \end{bmatrix} \text{ corresponding to } \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix}. \tag{4.2}$$

We set $\gamma = 3, A = 1$, and the frequency ranges to $\xi, \eta \in [-1, 1]; \nu \in [-1/8, 1/8]$ and the datacube size to $512 \times 512 \times 64$, i.e., 64 temporal images of size $512 \times 512$. We set $\gamma = 3$ instead of $\gamma = 2$ here because we are generating spatio-temporal images and using a 3-dimensional filter [17], $\gamma = 2$ may be used to generate sets of images with only 2-dimensional filtering via GENOBSPowerLaw, with any sets of images being temporally uncorrelated. See Fig. 1 for the spatio-temporal Stokes corre-
We have derived and presented 3 algorithms for statistical scene generation for polarimetric (or multi-hyper-spectral) image sets. The third algorithm, \texttt{GENObsPowerLaw3}, allows for correlation between multiple sets to occur, i.e., correlation between $s_1$ and $s_2$ of a Stokes image or correlation between the red and blue channels of a color image set. The algorithms generate statistically accurate sample images to test imaging system performance, for families of power law power spectral distributions. The specific parameters of the power laws can be specified for polarimetric data once future studies determine the parameters empirically, but a reasonable assumption is to use the parameters similar to those derived from color images. We present an example using our image generation algorithms, and elucidate the difference between spatial only power law distributions and spatio-temporal power law distributions.

FUNDING

Asian Office of Aerospace Research and Development (FA2386-15-1-4098).

REFERENCES

1. D. W. Dong and J. J. Atick, Network: Computation in Neural Systems 6, 345 (1995).
2. V. A. Van der Schaaf and J. v. van Hateren, Vision research 36, 2759 (1996).
3. A. Torralba and A. Oliva, Network: computation in neural systems 14, 391 (2003).
4. A. Srivastava, A. B. Lee, E. P. Simoncelli, and S.-C. Zhu, Journal of mathematical imaging and vision 18, 17 (2003).
5. I. J. Vaughn, O. G. Rodríguez-Herrera, M. Xu, and J. S. Tyo, 9613 (2015).
6. I. J. Vaughn, O. G. Rodríguez-Herrera, M. Xu, and J. S. Tyo, Proc. SPIE 9613 (2015).
7. I. J. Vaughn, A. S. Alenin, and J. S. Tyo, pp. 83640S–83640S (2016).
8. A. S. Alenin, I. J. Vaughn, and J. S. Tyo, Optics Letters 42, 458 (2017).
9. I. J. Vaughn, A. S. Alenin, and J. S. Tyo, Optics Express 25, 11954 (2017).
10. D. G. Childers and S. B. Kesler, , Modern spectrum analysis, vol. 331 (IEEE press New York, 1978).
11. G. Deodatis, Journal of engineering mechanics 122, 778 (1996).
12. P. M. Broersen and S. De Waele, IEEE Transactions on Instrumentation and Measurement 52, 1061 (2003).
13. N. Ashby, arXiv preprint arXiv:1103.5062 (2011).
14. N. J. Kasdin, Proceedings of the IEEE 83, 802 (1995).
15. H. H. Barrett and K. J. Myers, , Foundations of Image Science (Wiley-Interscience, Hoboken, New Jersey, United States, 2004).
16. F. Oberhettinger, , Fourier transforms of distributions and their inverses: a collection of tables, vol. 16 (Academic press, 2014).
17. J. B. Bassingthwaighte and G. M. Raymond, Annals of biomedical engineering 23, 491 (1995).
18. C. Heneghan, S. Lowen, and M. C. Teich, , pp. 213–217 (1996).