SILO-G: A Machine Learning Data Generator for Synthetic Ground-Based Observations of LEO Satellites

Nicole Gagnier, Trent Kyono, Jacob Lucas, Michael Werth
The Boeing Company
Justin Fletcher
Odyssey Consulting
Ian McQuaid, Michael Brannon
Air Force Research Laboratory

ABSTRACT

Numerous applications of machine learning for space domain awareness (SDA) have underscored the need for robust data sets of space objects, particularly Low Earth Orbit (LEO) satellites. In this paper, we present the Simulated Images of LEO Objects Generator (SILO-G), a lightweight data generator that uses satellite models and wave-optics simulations to create images of LEO satellites as if observed from a large ground-based optical observatory with varied turbulence conditions. SILO-G utilizes a graphical processing unit (GPU) based wave optics simulation to greatly increase wave optics simulation speed relative to CPU-bound approaches. The data generator framework of SILO-G enables a lightweight implementation in which the user can quickly edit parameters and generate large data sets without large storage requirements or processing overhead. Finally, we discuss some of the many novel applications that the SILO-G data set enables.

1. INTRODUCTION

The growing number of potential applications of machine learning to space domain awareness (SDA) underscores the need for robust and efficient camera image training data solutions. Previous work to create such a data set included the Scored Images of LEO Objects (SILO) data set. The SILO data set contains 90,000 simulated images of various resident space objects (RSOs) observed from the Advanced Electro-Optical System (AEOS) telescope on the summit of Haleakala in different poses and with various r_0 values [1]. The SILO data set has enabled many machine learning experiments for SDA [2, 3, 4, 5, 6] but the dataset possesses a number of limitations. All images are pregenerated using a massive quantity of computational hours on super computing resources, therefore generating new images to accommodate different experiment requirements is prohibitively difficult and time-consuming. These limitations provided the motivation to develop a lightweight, adaptable version of the SILO data set to expand upon the utility.

We present the Simulated Images of LEO Objects Generator (SILO-G). SILO-G combines pregenerated renders and pregenerated point spread functions (PSFs) into simulated camera images using a Python data generator. Furthermore, SILO-G includes a complementary Python wrapper for the open-source Multi-Threaded Adaptive Optics Simulator (MAOS) developed for the Thirty Meter Telescope (TMT) [7]. MAOS generates high-fidelity point spread functions using a Graphical Processing Unit (GPU), greatly increasing overall simulation speed to generate large numbers of PSFs. In addition, utilizing the SILO-G wrapper for MAOS affords the user control over the PSF properties not previously possible in the original fixed SILO data set. After convolving the pregenerated renders and PSFs, SILO-G applies user-specified camera effects and augmentations to produce fully simulated camera images as shown in Figure 1.

In this paper, we present the current SILO-G framework and discuss the image generation process. We then present some of the key performance, ease-of-use, and applicability improvements afforded by this second generation of SILO. Finally, we discuss future work with SILO-G including both experimental applications and planned enhancements to SILO-G.

DISTRIBUTION A. Approved for public release: distribution is unlimited.
Public Affairs release approval #OPS-20-38896

Copyright © 2020 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

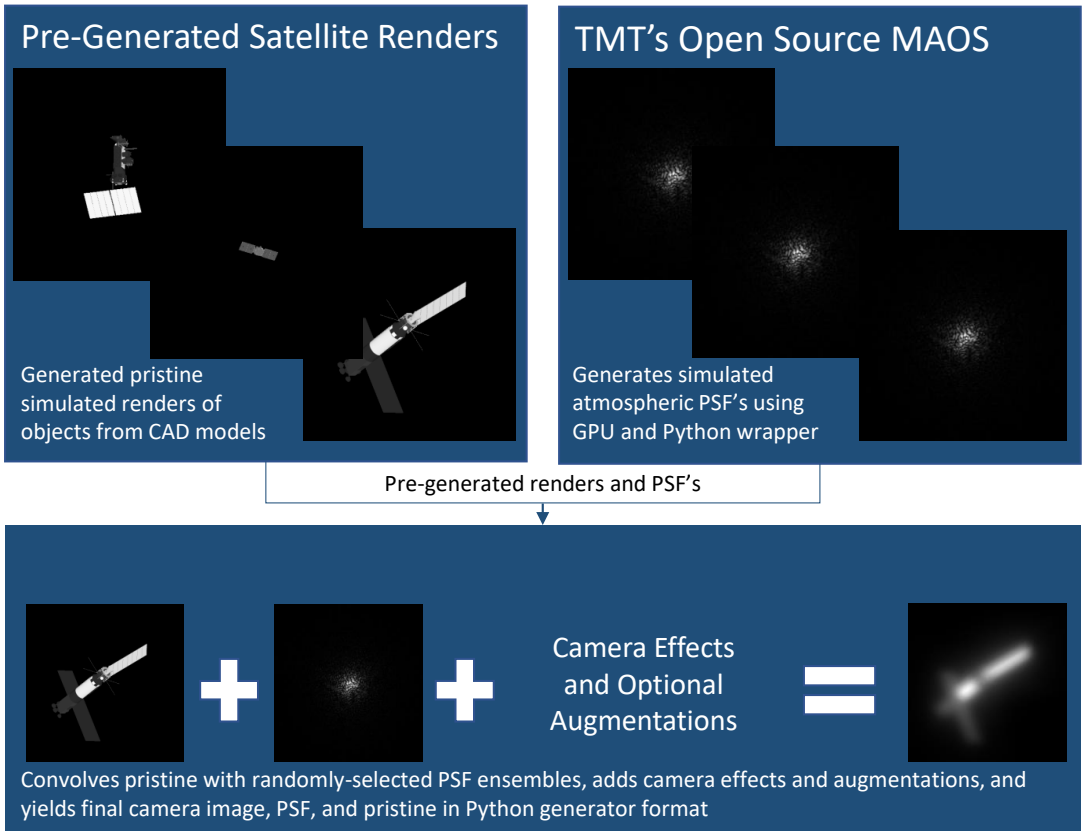


Fig. 1: SILO-G provides a pipeline for combining pregenerated satellite renders with PSFs generated through the accompanying MAOS wrapper to create simulated camera images in an efficient and user-friendly Python generator format.

DISTRIBUTION A. Approved for public release: distribution is unlimited.
Public Affairs release approval #OPS-20-38896

Copyright © 2020 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

2. SILO-G FRAMEWORK

SILO-G comprises a separate set of functions that initiate a MAOS session to generate the PSF diversity desired by the user and the main generator portion of the software. The MAOS portion creates a PSF library which is used along with a render library to simulate images. By separating the MAOS session from the generator, the generator remains lightweight and able to quickly produce large numbers of simulated images while training.

The MAOS session functionality allows the user to input lists of values for each of the parameters used in creating the PSFs. The session will then include all possible combinations of the various input parameters provided by the user. Each MAOS output file contains the user-specified number of frames and is converted automatically from binary to a FITS [8] file format. Each MAOS output file corresponds to one unique combination of the input parameters. A comma separated value (CSV) key listing each file and its corresponding run parameters is produced at the end of the full session.

With directories of PSFs and pristine renders ready, the user then runs the main portion of SILO-G, the generator function. Since MAOS produces simulated PSFs with 1/800 second integration times, SILO-G sums frames to reach the desired final camera integration time. As shown in Figure 2, the generator convolves randomly selected PSFs with the pristine renders. SILO-G then conducts all necessary scaling to achieve the desired instantaneous field of view (IFOV) and summation to reach the desired integration time, after which any camera effects and/or augmentations are applied. Through this framework, the user can generate an arbitrarily large number of simulated images in a format conducive to network training.

3. IMAGE GENERATION

The SILO-G package produces images by using a combination of pregenerated pristine satellite renders and pregenerated PSF files. This is accomplished with the MAOS session software in two main steps: PSF formation and camera simulation.

3.1 PSF Formation using MAOS

We utilize the open-source software MAOS, developed for the TMT program, as the atmospheric turbulence simulator for generating PSFs. MAOS is able to run on GPU hardware for performance benefits and is written in C. MAOS utilizes a series of configuration files to specify parameters ranging from turbulence profile to telescope design and sizing [7]. The Python wrapper included in SILO-G exposes a subset of the many parameters of MAOS to make interacting with the software easier for the average user. Additionally, the Python wrapper can accept a list as input for each of the parameters, causing MAOS to execute every possible parameter combination to produce a wide data set of PSFs. Some of the main input parameters and corresponding defaults are shown in Table 1.

3.2 Camera Simulation

SILO-G applies the series operations depicted in Figure 2, creating simulated ground-based low earth orbit (LEO) observation images from pristine renders and PSFs. MAOS' default simulation time step is 1/800s so SILO-G sums a series of frames to form the full camera integration time. Thus, each iteration of the generator randomly selects a file and a starting frame index to increase variability over the data generation period. SILO-G randomly selects a PSF file, normalizes the PSFs contained in the file, then generates a random index for the first frame to be used in summation. The PSFs within the selected file are normalized to one and SILO-G selects a subset of PSF frames that will sum to the camera integration time using the random starting index. SILO-G convolves each of the subset of PSF frames with the randomly selected pristine render and then sums the resulting images until the camera integration time is reached. Transmission losses, Poisson noise, quantum efficiency effect, and Gaussian read noise based on the user input parameters for these values as described in Table 1 are applied to the summed image. Once the camera effects are applied, the generator yields an image which can then be optionally modified using standard machine learning augmentations. SILO-G will iterate through each PSF file using different indices for start frames until all of the pristine renders have been utilized and then will continue on to selecting a completely new PSF file, beginning the process again. This process is summarized in Algorithm 1.

DISTRIBUTION A. Approved for public release: distribution is unlimited.
Public Affairs release approval #OPS-20-38896

Copyright © 2020 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

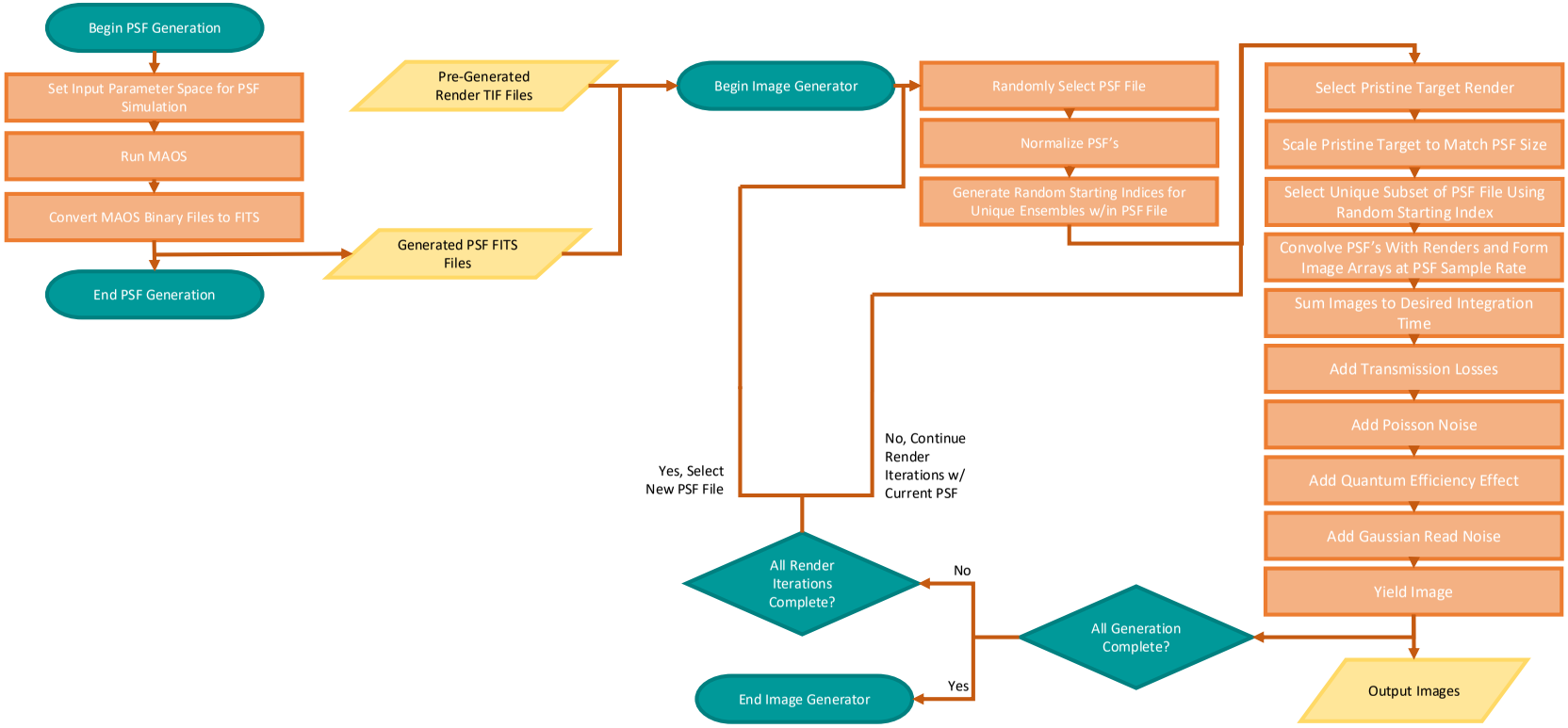
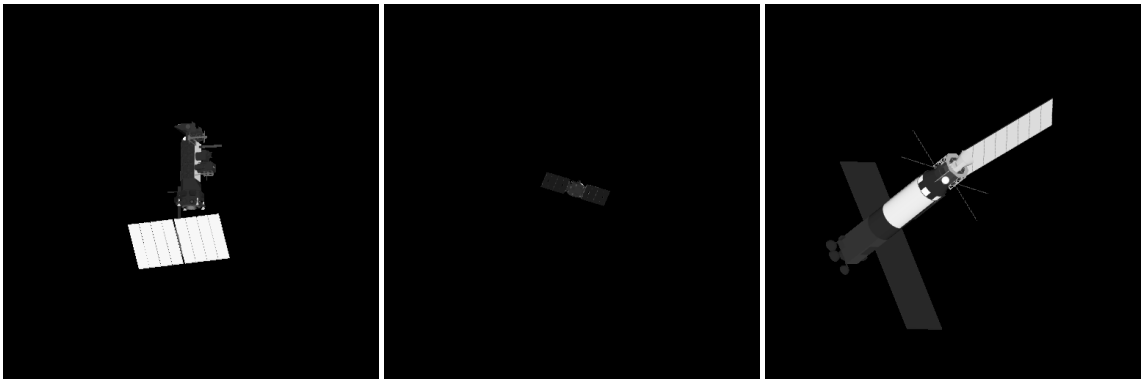


Fig. 2: Schematic of SILO-G implementation. pregenerated renders are convolved with MAOS-generated PSFs and undergo a series of operations to simulate final camera images.

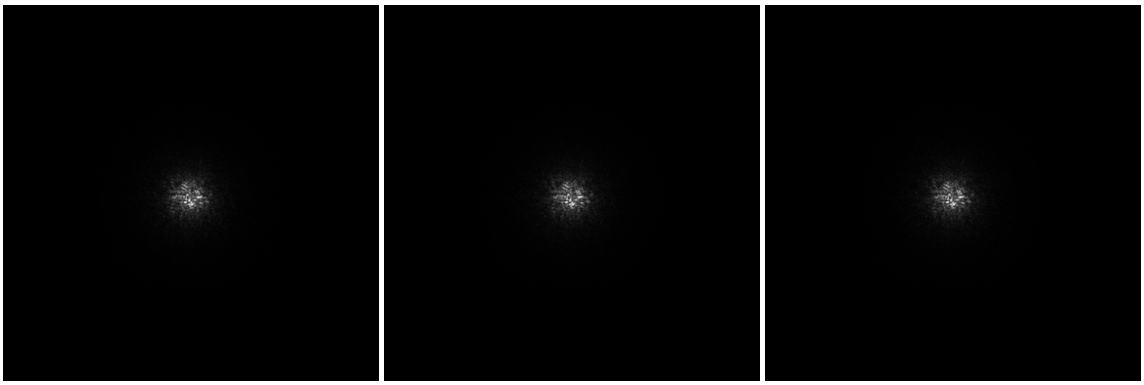
DISTRIBUTION A. Approved for public release: distribution is unlimited.
Public Affairs release approval #OPS-20-38896

Copyright © 2020 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com



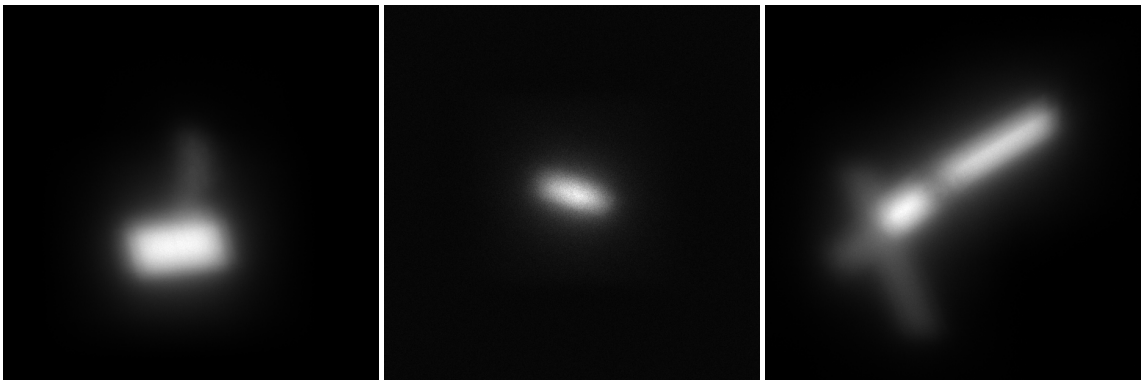
(a) (b) (c)

Fig. 3: Pristine images of (a) NOAA 11 (SCN: 19531) (b) MIGHTYSAT 2 (SCN: 26414) (c) ASTEX 1 (SCN: 5560) satellites.



(a) (b) (c)

Fig. 4: Images (a), (b) and (c) show examples of MAOS generated PSFs.



(a) (b) (c)

Fig. 5: Simulated camera images of (a) NOAA 11 (SCN: 19531) (b) MIGHTYSAT 2 (SCN: 26414) (c) ASTEX 1 (SCN: 5560) satellites using the SILO-G pipeline.

DISTRIBUTION A. Approved for public release: distribution is unlimited.
Public Affairs release approval #OPS-20-38896

Copyright © 2020 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

Table 1: SILO-G Example Parameters

SILO-G Parameters

Parameter	Description	Default Value
PSFs to Sum	Number of PSFs to sum to make one full camera integration time.	20
Render IFOV	Instantaneous field of view (IFOV) of input renders in radians/pixel.	[100e-9, 100e-9]
Camera IFOV	Instantaneous field of view (IFOV) of desired output camera image in radians/pixel.	[100e-9, 100e-9]
Camera Size	Desired output camera image size in pixels.	[256, 256]
Read Noise	Camera read noise in ADU.	4
Transmission Coefficient	Optical transmission coefficient	0.7
Quantum Efficiency	Camera quantum efficiency	0.9

MAOS Session Parameters

Parameter	Description	Default Value
r_0 at Zenith	Fried parameter r_0 at given zenith angle.	0.247
Zenith Angle	Observation angle between zenith and the PSF in degrees.	45
Seeds	A list of random number generator seeds for simulation.	1
PSF Size	Output PSF size in pixels.	[256, 256]
Number of Frames	Number of PSF frames to produce per run.	50
Wavelength	Observation wavelength in meters.	750e-9

4. PERFORMANCE IMPROVEMENTS

The most striking improvement provided by SILO-G compared to the original SILO data set is the vastly decreased computation time required to produce images. The original SILO project produced 11 million 256x256 images in 500,000 hours of compute time. Figure 6 shows the compute time required for SILO vs. SILO-G using the same 256x256 image and PSF size (generation of which is included in the SILO-G compute time). In as much time to compute 60,000 images using SILO-G, SILO produces under 10,000.

In addition to the computation time improvements, whereas the SILO data set consists of images pregenerated with a fixed set of parameters, SILO-G allows the user to tailor the simulations using a number of input variables, such as camera characteristics, turbulence parameters, and choice of satellites renders. The framework is designed to allow the user to easily adapt the generated data to fit their needs and also add on any modifications, if necessary. SILO-G provides a lightweight data solution applicable to many users interested in generating LEO data sets for SDA machine learning applications.

5. CONCLUSION

We have presented SILO-G, a machine learning data generator for ground-based observations of LEO satellites. SILO-G addresses some of the shortcomings of the original SILO data set and is designed to allow for wider-use and applicability in addition to incorporating significant performance gains. While SILO-G has already greatly increased our ability to quickly generate diverse training data for SDA applications, we plan to expand upon its functionality in future work. First, we plan to transition the convolution and camera simulation computation from the CPU to the GPU in order to further increase overall simulation speed. Second, we plan to include 1:1 mapping of the pristine render and the PSF to the degraded image to facilitate new experiments where neural networks are trained to estimate the PSF in a degraded image. For example, we can leverage the PSF to further enhance machine learning, by obligating learned

DISTRIBUTION A. Approved for public release: distribution is unlimited.
Public Affairs release approval #OPS-20-38896

Copyright © 2020 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

Algorithm 1: SILO-G Image Formation

Data: m Input Renders, n Input PSFs
Result: Simulated Camera Images
while *Images are requested* **do**
  Select random PSF file;
  Normalize PSFs in file;
  Generate list of n random start indices for PSF subsets;
  for *i=0:n pristine renders* **do**
    Scale pristine image to match PSF size;
    Select starting PSF frame index for PSF subset from list, i.e. k=indexList[i];
    for $j = 0$*:Number of PSFs to sum to reach desired integration time* **do**
      Form camera image with transfer functions and convolutions using pristine render and PSF frame at index $k + j$;
      Scale image to desired camera image size;
    end
    Sum images to desired integration time;
    Add transmission losses;
    Add Poisson noise (shot noise);
    Add quantum efficieny effects;
    Add Gaussian noise (read noise);
    Yield image;
  end
end

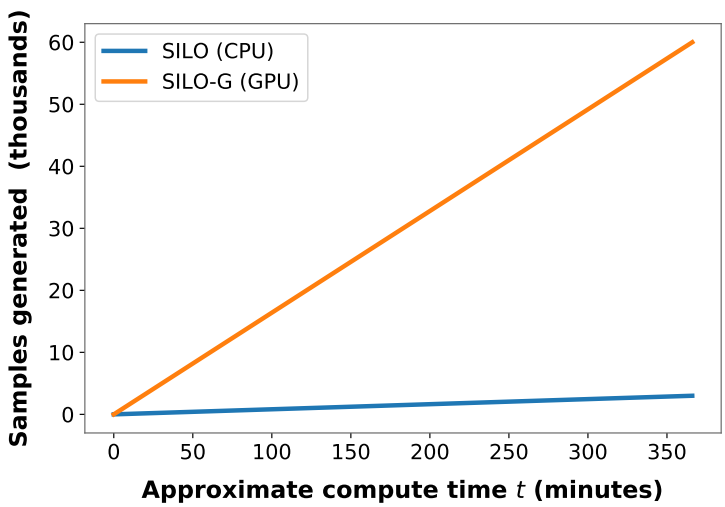


Fig. 6: Plot showing SILO compute time (Intel Xeon Platinum 8168 CPU) vs. SILO-G compute time (GPU) using 256x256 images and 200 frame pregenerated PSF ensembles with 12 CPU cores (Intel i9-9820X), 64 GB of RAM, and 2 NVidia RTX 2080 Ti graphics cards.

DISTRIBUTION A. Approved for public release: distribution is unlimited.
Public Affairs release approval #OPS-20-38896

Copyright © 2020 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

representation towards true or representative PSFs. We plan to release SILO-G via the Air Force's Unified Data Library (UDL) in the coming months. We hope that SILO-G and its future development will provide the SDA community with a user-friendly, lightweight data generation solution for simulated images of ground-based observations of LEO satellites.

REFERENCES

[1] Michael Werth, Jacob Lucas, Trent Kyono, Ian McQuaid, and Justin Fletcher (in press). SILO: A machine learning dataset of synthetic ground-based observations of LEO satellites. In *IEEE Aerospace*. 2020.

[2] Trent Kyono, Jacob Lucas, Michael Werth, Ian McQuaid, and Justin Fletcher (in press). Determining multi-frame blind deconvolution resolvability using deep learning. In *Advanced Maui Optical and Space Surveillance (AMOS) Technologies Conference*. 2019.

[3] Michael Werth, Jacob Lucas, Trent Kyono, Ian McQuaid, and Justin Fletcher (in press). Quality-weighted iterative deep convolution (QWID). In *Advanced Maui Optical and Space Surveillance (AMOS) Technologies Conference*. 2020.

[4] Jacob Lucas, Trent Kyono, Ian McQuaid, and Justin Fletcher (in press). Deep learning approach for satellite orientation determination from ground based resolved images. In *Advanced Maui Optical and Space Surveillance (AMOS) Technologies Conference*. 2020.

[5] Jacob Lucas, Michael Werth, Trent Kyono, Ian McQuaid, and Justin Fletcher. Automated interpretability scoring of ground-based observations of LEO objects with deep learning. In *IEEE Aerospace*. 2020.

[6] Trent Kyono, Jacob Lucas, Michael Werth, Brandoch Calef, Ian McQuaid, and Justin Fletcher. Machine learning for quality assessment of ground-based optical images of satellites. *Optical Engineering*, 59(5):1 – 15, 2020.

[7] Lianqi Wang and Brent Ellerbroek. Fast end-to-end multi-conjugate AO simulations using graphical processing units and the MAOS simulation code. 01 2011.

[8] Pence, W. D., Chiappetti, L., Page, C. G., Shaw, R. A., and Stobie, E. Definition of the flexible image transport system (fits), version 3.0. *A&A*, 524:A42, Nov 2010.

DISTRIBUTION A. Approved for public release: distribution is unlimited.
Public Affairs release approval #OPS-20-38896

Copyright © 2020 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com