Estimating Satellite Orientation through Turbulence with Deep Learning

Jacob Lucas, Trent Kyono, Michael Werth, Nicole Gagnier, Zackary Endsley
The Boeing Company
Justin Fletcher
Odyssey Systems Consulting
Ian McQuaid
AFRL

Abstract

Determination of satellite orientation is critical for evaluating spacecraft health and characterizing motion. Ground based sensors are under utilized for this task due to the image degradation imposed by atmospheric turbulence. We propose a deep learning based approach to real-time pose estimation from ground based sensor images. We leverage an extensive collection of simulated imagery and atmospheric turbulence to both train a pose estimator and to characterize performance under a variety of real scenarios. In addition, we show that for deeply degraded conditions where pose estimator performance suffers, the application of an image recovery network can restore accuracy without impacting real-time status.

1. Introduction

This paper examines the applicability of a straightforward deep learning approach to satellite pose estimation based on ground based images of LEO (Low Earth Orbit) satellites. Similar efforts have been succesful on RGB images of objects such as cars [1] and faces [2], however most of this work assumes large, clear images. Astronomical sensors typically have far fewer pixels than consumer cameras, and due to the light starved nature of astronomy the signal to noise ratio (SNR) can be very low. Additionally, atmospheric turbulence guarantees the images will be degraded. Any pose-estimation-from-images approach must take into account all of these potential degraders to be viable.

Here we have simulated a substantial set of 216,000 pristine images of unique Hubble Space Telescope (HST) poses as they would appear viewed in the absence of any atmosphere from the AEOS 3.6m telescope on Haleakala. We have developed an in-house pipeline to apply realistically simulated arbitrary atmospheric conditions to renders, which takes into account both atmospheric and camera centric degraders. We examine the upper performance limit for our chosen network architecture by first training and validating on the pristine images. We then show how performance degrades in the presence of atmosphere, as well as how it is affected by image resolution. Lastly we present a potential solution for improving inference accuracy on degraded images.

We provide a discussion of related works in Section 2. Section 3 details a formalization of our problem and approach. Section 4 provides a discussion of our dataset, training architecture, hyperparameters, and our experimental results. We conclude with brief remarks in Section 5.

DISTRIBUTION A. Approved for public release: distribution is unlimited.
Public Affairs release approval #AFMC-2020-0425

2. Related Works

Several related works have explored the applications of CNNs and deep learning to astronomy. For noise reduction, [3] recently presented a proof-of-concept neural network for denoising the bispectrum for astronomical image recovery on synthetic data. For classification, [4, 5, 6] investigated the application of object classification using neural networks on photometric light curves and showed promising results. Using Generative Adversarial Networks (GANs), [7] recovered features from artificially degraded images with worse seeing and higher noise than the original with a performance that far exceeded the capabilities of simple deconvolution. Additionally, [8] used a GAN to generate more realistic images of galaxies than existing state of the art. [9] used machine learning to automatically segment and label galaxies in astronomical images. [10] showed promising results using an autoencoder for real-time MFBD of solar images. The success of these approaches motivates our application of similar networks to the pose angle estimation of LEO (Low Earth Orbit) observations.

Throughout industry and academia, CNNs for pose-estimation is a rapidly growing topic and has countless applications. One of the primary topics in the literature is human-pose estimation, which we refer the reader to the survey paper of [11]. This problem of human-pose estimation (and related applications) is of non-rigid objects that are not statically defined. In this work we are focusing on rigid objects that do not have any moving parts (and are non-pliable) and are thus static in 3D form. This is a commonly addressed problem in robotic vision systems. [12] Explores rigid body pose estimation based on the Lagrange–d'Alembert principle and variational mechanics without needing a 3D model. A work similar to ours is PoseCNN [13] which uses a CNN to regress the translation and rotation matrices assuming a static 3D model.

3. Rigid-body Angle Estimation

The problem of 6D pose estimation is to estimate the transformation of a rigid body from the object coordinate system O to the camera coordinate system C given an input image. The rigid transformation is comprised of a transform containing a 3D rotation R, which specify the rotation angles around X, Y, and Z axes of O, and a 3D translation T, where T is the coordinate of the origin of O in C. From the stand-point of ground-based imaging, it is safe to assume that the location of T is known via two-line element set (TLE) or state-vector which can be known ahead of time or recovered from tracking the object. Because of this, in this work we only focus on estimating the true rotation R of the object of interest. Further assumptions include: (i) a single input camera image, (ii) the 3D model of the object is available, and (iii) the object coordinate system is defined in the 3D space of the model.

Here we describe our CNN approach to rigid-body angle estimation. Let I be a 2-dimensional image (matrix) with m rows and n columns that belongs to the vector space $\mathcal{I} \in \mathbb{R}^{m \times n}$, let the rotation R expressed as a quaternion belonging to the vector space $\mathcal{R} \in \mathbb{R}^4$, and let Q be a mapping function, such that $Q : \mathcal{I} \rightarrow R$. In this work rather than regressing R as a standard rotation matrix, we focus on regressing the quaternion representation of R, which provides a continuous representation of the object with no discontinuities. The quaternion can be defined as an alternative to R and is generally represented as $a + bi + cj + dk$, where a, b, c and d are real numbers and i, j, k are quaternion units.

DISTRIBUTION A. Approved for public release: distribution is unlimited.
Public Affairs release approval #AFMC-2020-0425

Table 1: Parameters used for image generation.

Parameter	Values	Units
Image Size	256x256	pixels
IFOV	100	nrad
PSF Generation Rate	800	Hz
Frame Rate	40	Hz
Waveband	Bessel I	-
Turbulence Profile	MK50P	-
Fried Parameter r_0	10,15,25,40,80	centimeters

4. Experiments

This section briefly covers the data used for this work, our experimental settings (training architecture and regimes), and our experimental results.

4.1. Datasets

This study utilized a supervised training methodology, and as such truth training data was required. This data consisted of multiple poses of the Hubble Space Telescope (HST) rendered as if viewed (without any degradation from atmosphere or other sources) from the Advanced Electro-Optical System (AEOS) at the summit of Haleakala. A total of 216,000 renders were produced, all with different poses relative to the observer. This render set is referred to as the pristine set, since it has no degradation applied.

To better characterize real-world performance, Point Spread Function (PSF) ensembles were generated to represent a variety of viewing conditions from AEOS. Each ensemble was characterized by its Fried Parameter, r_0. The ensembles were generated using the Multi-threaded Adaptive Optics Simulator (MAOS) code package [14]. A summary of simulation parameters is in Table 1. Example images of an HST render at each of the chosen r_0 values are shown in Figure 1. The PSFs along with realistic camera noise were then applied to the pristine renders. This resulted in an additional five 216,000 image training sets representing reasonable AEOS seeing conditions (Table 2) referred to as the degraded data sets. To create what we refer to as the reconstructed data sets we applied a Generative Adversarial Network (GAN) trained to recover LEO satellite images through turbulence to the degraded image sets shown in 5 with the goal of improving image quality while risking the loss of some details due to network inference error. Finally, In order to simulate sensors with less resolving power, a pixel binning approach was employed. This used the pristine data set and 'binned' pixels in order to reduce effective IFOV.

4.2. Training Methods and Performance

In this work we began with the pre-trained ImageNet models ResNet50, VGG16, InceptionV3, and Xception. All tested networks had their top layers replaced by a global average pooling layer followed by a set of three dense and batch normalization layers. The final dense layer having 4 neurons and a sigmoid activation function to match the dimensionality and scale of a quaternion prediction. A brief trade was done to determine

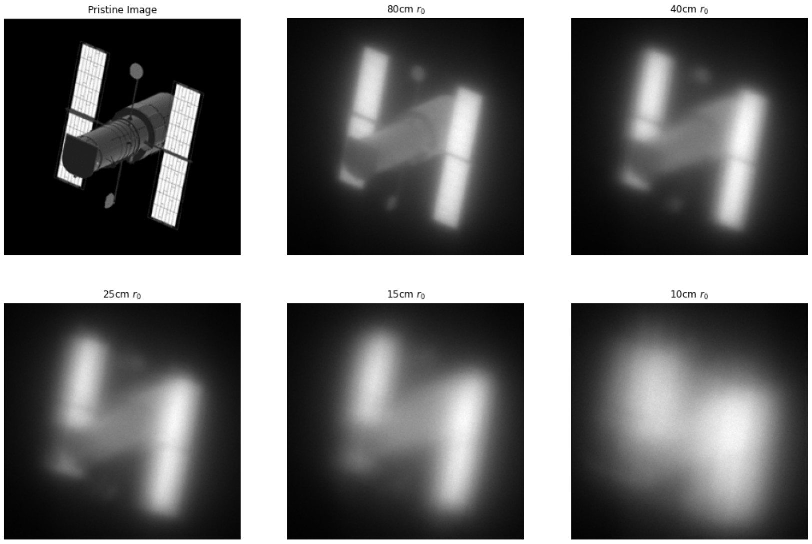

Figure 1: Example images representing various seeing conditions.

Table 2: Representative Seeing Conditions for Haleakala

Seeing Condition	r_0 (cm)
Poor	10
Average	15
Good	25
Exceptional	40
Typical AO	80

DISTRIBUTION A. Approved for public release: distribution is unlimited.
Public Affairs release approval #AFMC-2020-0425

Table 3: MAE of direct quaternion regression *(q1 through q4)*.

Imagenet Backbone	Validation MAE	Image Quality
VGG16	0.164	Pristine
ResNet50	0.038	Pristine
InceptionV3	0.091	Pristine
Xception	0.030	Pristine

Table 4: MAE of direct quaternion regression vs angular resolution

Imagenet Backbone	Validation MAE	IFOV	Image Size
Xception	0.030	100nrad	256x256
Xception	0.035	200nrad	128x128
Xception	0.122	400nrad	64x64

which modified network performed best with the pristine data set, and Xception was chosen as the most performant, as can be seen in Table 3. Figure 2 presents the error of this network on pristine (ideal) images in degrees after conversion to Euler angles. The network trade was accomplished by training each network in an iterative manner. First freezing all base layers and training only the top for 15 epochs, then freezing top layers and training only the base for 40 epochs. This was repeated for a total of two iterations, and then finally the complete model was trained for 50 epochs. Models were saved every epoch. The Mean Absolute Error (MAE) for quaternion regression is not intuitive, but is adequate for comparing performance; it is shown in Tables 3 and 4. Errors were calculated on a set of 5000 test images of poses that had not been previously seen by the network.

Smaller images use less memory and have a reduced processing time, inspiring an addition trade comparing image size. It is obvious that image size could have an effect on pose estimation. If size is reduced and IFOV held constant, then portions of the object may extend beyond the field of view. If size is reduced and full FOV held constant, then resolving power is reduced. A short trade was done to try to quantify the effect on performance of reducing the image size while retaining the full FOV. The results are shown in Table 4. Based on this it was determined that the remainder of this study would use full size 256x256 renders.

A training procedure identical to that used in the network trade was then performed using Xception as the network. A separate Xception model was trained for each of the 5 different degradation cases represented in Table 2. The center column of Table 5 illustrates the relative performance across the different seeing conditions.

In order to determine any improvements gained by applying the reconstruction GAN we trained a separate Xception model for each of the 5 reconstructed data sets. A summary of the performance of these networks is shown in the right most column of Table 5. The reconstruction network performs poorly for r_0 values less than 25cm.

It is interesting to see that networks trained on this reconstructed data are almost universally less accurate. This was not the expected result and is likely due to background artifacts and possible hallucinations intro-

DISTRIBUTION A. Approved for public release: distribution is unlimited.
Public Affairs release approval #AFMC-2020-0425

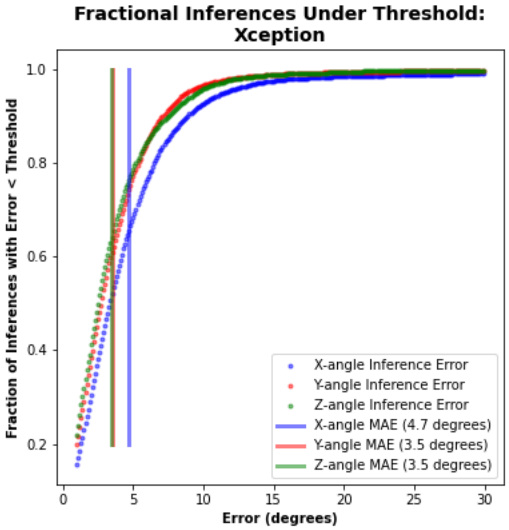

Figure 2: Fractional inferences under different thresholds. Vertical bars represent the MAE values for each axis. The combined 3-Axis MAE for this network was 5.6 degrees

Table 5: Optimal performance at each turbulence level before and after neural network reconstruction. Each entry represents the 3-Axis MAE of a network trained to recover pose for that specific scenario.

r_0 (cm)	MAE Degraded (degrees)	MAE Reconstructed (degrees)
10	18.4	29.1
15	15.5	21.9
25	13.4	16.9
40	13.4	13.3
80	11.6	15.3

Table 6: Performance of network trained only on Pristine data when tested on degraded and reconstructed data sets. Shown as 3-axis MAE. For comparison, this network performed to 5.6 degrees 3-Axis MAE on pristine data.

r_0 (cm)	MAE Degraded (degrees)	MAE Reconstructed (degrees)
10	68.6	74.3
15	63.3	64.7
25	54.1	50.2
40	39.7	33.4
80	21.7	17.7

DISTRIBUTION A. Approved for public release: distribution is unlimited.
Public Affairs release approval #AFMC-2020-0425

duced by the GAN. The similar performance for the 40cm and 80cm r_0 reconstructed data implies some introduced accuracy bound. Table 6 illustrates the potential utility of the reconstruction GAN. Here the pristine network (Xception trained on only pristine renders) performance was measured for all 10 data sets. While performance is poor in all cases, it is improved by the reconstruction network for 25cm and better seeing conditions. Table 6 implies that applying a reconstruction network can indeed improve a pose estimation network, the results in Table 5 suggest our reconstruction network was simply not good enough. Repeating this study with MFBD (Multi Frame Blind Deconvolution) or some other established algorithm in lieu of the GAN used here could provide a performance baseline, but is outside the scope of this study. Additionally, the pristine network inference accuracy could be used as a scoring metric for future reconstruction GAN training to assist in reducing hallucination and improving overall quality. Another result of note was the performance plateau seen in Table 5, where from 80-25cm r_0 the performance remains roughly the same. The implication is that the increased turbulence does not appreciably tax the network. This could significantly improve the utility of a ground based approach, as AO or post processing would not be required. The performance seen in Table 5 suggests that further effort into improving accuracy is warranted, as the seeing conditions required are not prohibitive, and image reconstruction may not be necessary.

4.3. Reproducibility

For reproduction, all neural networks were trained using Python 3 and TensorFlow 2.0 [15]. Operating system and hardware specifications include RedHat Linux 7 on an NVidia DGX Workstation with four Tesla V100 GPUs with 32 GB of memory on each card. Because the goal of this work was a feasibility investigation, we did not tune or search for optimal hyperparameters.

5. Conclusion

We have shown that with thorough training data a CNN can do an admirable job of pose estimation from a single image. We have also shown how these estimates suffer under real world conditions and illustrated how this performance could be bolstered via application of additional machine learning approaches. For future work we plan to apply an improved reconstruction network trained on a more extensive data set along with a secondary fine tuning network to further increase accuracy. We will also be testing different training routines, along with hyperparameter tuning. Finally, we will be incorporating real data collected with AEOS.

References

[1] S. Mahendran, H. Ali, and R. Vidal. 3d pose regression using convolutional neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 494–495, 2017.

[2] Zhaoxiang Liu, Zezhou Chen, Jinqiang Bai, Shaohua Li, and Shiguo Lian. Facial pose estimation by deep learning from label distributions, 2019.

[3] Jacob Lucas, Brandoch Calef, and Trent Kyono. Recovering astronomical images with deep neural network supported bispectrum processing. In *Advanced Maui Optical and Space Surveillance (AMOS) Technologies Conference*. 2018.

[4] B. Jia, K. D. Pham, E. Blasch, Z. Wang, D. Shen, and G. Chen. Space object classification using deep neural networks. In *2018 IEEE Aerospace Conference*, pages 1–8, March 2018.

[5] Richard Linares. Space object classification using deep convolutional neural networks. 07 2016.

[6] Robert Furfaro, Richard Linares, and Vishnu Reddy. Space objects classification via light-curve measurements: Deep convolutional neural networks and model-based transfer learning. 09 2018.

[7] Kevin Schawinski, Ce Zhang, Hantian Zhang, Lucas Fowler, and Gokula Krishnan Santhanam. Generative adversarial networks recover features in astrophysical images of galaxies beyond the deconvolution limit. *Monthly Notices of the Royal Astronomical Society: Letters*, 467(1):L110–L114, 01 2017.

[8] Levi Fussell and Ben Moews. Forging new worlds: high-resolution synthetic galaxies with chained generative adversarial networks. *Monthly Notices of the Royal Astronomical Society*, 485(3):3203–3214, 03 2019.

[9] Alex Hocking, James E. Geach, Yi Sun, and Neil Davey. An automatic taxonomy of galaxy morphology using unsupervised machine learning. *Monthly Notices of the Royal Astronomical Society*, 473(1):1108–1129, 09 2017.

[10] A. Asensio Ramos, J. de la Cruz Rodriguez, and A Pastor Yabar. Real-time multiframe blind deconvolution of solar images. 06 2018.

[11] Yucheng Chen, Yingli Tian, and Mingyi He. Monocular human pose estimation: A survey of deep learning-based methods. *Computer Vision and Image Understanding*, 192:102897, 2020.

[12] Maziar Izadi and Amit K. Sanyal. Rigid body pose estimation based on the lagrange–d'alembert principle. *Automatica*, 71:78 – 88, 2016.

[13] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. 2018.

[14] Lianqi Wang. Multi-threaded adaptive optics simulator. `http://lianqiw.github.io/maos/`, 2019.

[15] Martín Abadi et.al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

DISTRIBUTION A. Approved for public release: distribution is unlimited.
Public Affairs release approval #AFMC-2020-0425