2023 IEEE International Conference on Robotics and Automation (ICRA)
Workshop on Active Methods in Autonomous Navigation
29 May – 2 June, 2023, London, UK

Lost in Space: Visual Satellite Navigation

Kyle McCleary[1], Paulo R.M. Fisch[1], Saral Tayal[1], Zac Manchester[1], and Brandon Lucia[1]

***Abstract*— The large-scale proliferation of nanosatellites has led to a need for a low-cost, timely, and effective method for orbit determination (OD). In this work, we develop visual nanosatellite OD using a low-cost onboard camera and onboard processing as inputs to an OD filter. We achieve similar accuracy to ground radar (on the order of 100 m), but much faster.**

I. Introduction

As the proliferation of nanosatellites continues and constellation sizes grow, locating and tracking tiny satellites, is a key requirement. Nanosatellites have many applications, including disaster response [1], agriculture [2], and land use classification [3]. These applications require knowledge of a satellite's orbit to allow communication to the ground. Existing techniques for nanosatellite orbit determination (OD) are impractically expensive or imprecise, relying on expensive, power-hungry satellite components or complex, costly ground infrastructure.

The goal of this work is to make OD cheap and practical, using low-cost visual sensing and machine vision algorithms that are efficient enough to run onboard small satellites. We describe a system that models Earth's landmarks, trains a model to run onboard a satellite to identify those landmarks, and uses a straightforward state-estimation algorithm to localize a satellite. Our results show that automatically identifying landmarks on Earth is feasible. We show that training a machine learning (ML) model to find those landmarks and running that model on a nanosatellite is within the satellite's power, energy, and time constraints. We then show that we enable 100 m precision OD with no ground infrastructure, and with simple, low-cost, hardware on the satellite.

TABLE I
Comparison of Orbit Determination (OD) Methods

Property/Method	GPS OD	Ground Radar	Visual OD
Largest Dimension	71 mm [4]	Off Satellite	<50 mm
Mass	31 g [4]	Off Satellite	<15 g
Power	1-2 W [4]	Off Satellite	<1 W
Cost	∼$5k	∼$10k	<$100
OD Time	Secs [5]	Wks-Mos [6]	Secs-Hrs
Precision	1.5 m [5]	∼100 m	∼100 m

A. Current Post-Launch Satellite Location Methods

Existing techniques for post-launch satellite localization are costly and complex. Some satellites include a Global Positioning System (GPS) receiver for precise OD [7], [8], [9]. A GPS receiver usable in space at high speeds is expensive and large, which precludes use in very small satellites [10], [11]. Other satellite missions use terrestrial radar [8], [12] to locate the satellite after launch, which can sometimes take months, and may fail to ever uniquely identify the satellite [12]. Table I summarizes these existing methods.

[1]The authors are with Carnegie Mellon University, Pittsburgh, PA, 15213 USA. Email: {kmcclear, pfisch, stayal, zmanches, blucia}@andrew.cmu.edu

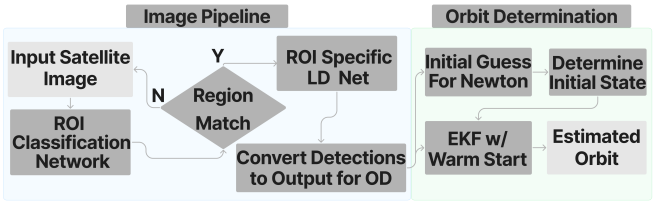

Fig. 1. Block diagram of end-to-end orbit determination pipeline.

B. Visual Orbit Determination

We develop visual OD using a low-cost onboard camera and on-satellite ML inference. Visual OD is simple and cheap. Modern cameras are available for two-to-three orders of magnitude cheaper than GPS receivers [13], [14], and recent work [15] showed the feasibility of onboard ML inference. The technique works by first capturing images and processing them to match automatically inferred landmarks on Earth (e.g., distinctive coastal regions). The technique then uses a standard Kalman filter to fully determine the orbit in a short period of time. Figure 1 shows an overview.

The main contribution of this paper is a novel, visual-only approach to OD using a combination of cheap cameras, on-satellite ML inference, and extended Kalman filtering. We hope to test the end-to-end initial OD process on orbit later in 2023.

II. Sattelite Image Geolocation

Visual OD begins with the image capture and processing pipeline shown in Figure 2. The image pipeline first classifies an image as a particular Region of Interest (ROI) on Earth. The pipeline then detects pre-defined Locations of Interest (LOIs) in the image. ROI classification uses a general network that classifies across all regions of Earth. LOI detection runs one of several ROI-specific detection networks to find LOIs. A series of LOIs can be used by an OD filter pipeline (described below) to determine the satellite's orbit.

A. Data Gathering

We developed our ROI and LOI models based on well-known ML model architectures and open data sources that we curated and combined to produce a useful system. We first describe the data we use to train our models.

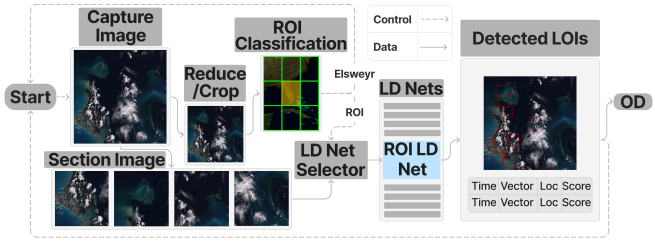

Fig. 2. Image pipeline. Captured images are cropped and fed into a Region of Interest (ROI) classification network. If classified as uninteresting, the pipeline returns to capture. Otherwise, the class flows to the Landmark Detection (LD) network selector. The original image is sectioned into smaller images, which flow to the LD network selector. The LD selector uses the ROI class to feed the images to the corresponding LD network. The output of the LD network is then passed to the OD filter.

We trained our models using Landsat [16] and MODIS [17] images downloaded from Google Earth Engine [18]. Using the specifications of the OV5640-C camera [14] (representative of a low-cost nanosatellite sensor), we calculate expected ground sample distance (GSD) and footprint of imagery captured at a 400 km altitude. The physical dimensions of the scene are derived using the focal plane dimensions, the focal length, and the distance to the scene. After computing a scene's dimensions we compute GSD.

To define classes for ROI classification, we perform a distinctness study using NASA's Blue Marble imagery [19]. It begins by taking a box from Blue Marble of a size from a representative orbit and camera combination (e.g., 400x300 km) and then shifting it pixel-by-pixel to find the greatest normalized correlation coefficient and the *second* greatest normalized correlation coefficient across Blue Marble. The difference between the correlation coefficients is the distinctness for that box. This process is repeated by selecting boxes originating from every pixel on Blue Marble. Then, the process is repeated again for boxes of a range of sizes. The distinctness for each pixel is recorded and summed during this process. The most distinct pixels are shown in Figure 3.

We then divide Earth into regions according to the Military Grid Reference System (MGRS) [20]: regions of roughly 6 degrees of longitude by 8 degrees of latitude. The most distinct regions from the distinctness study were used to select MGRS cells to act as classes for ROI classification. For training the ROI classification network, we collected 5 random 300x300 km MODIS images at 600 m/pixel from every day in 2020 over each ROI and labeled them using the corresponding MGRS cell. This ensured good coverage of weather, lighting, and seasonal variations, and led to a training set with over 100,000 images. Additionally, 20 random crops per day in 2020 were used for the "uninteresting" class. The validation set contains a random crop from every day in 2019 for each ROI from both Landsat and MODIS.

To define a set of LOIs to detect, we divide the NASA Blue Marble imagery at 500 m spatial resolution into the appropriate ROIs. We then use the OpenCV [21] implementation of the fine grained saliency approach [22] on each region to create a saliency map. Next, we select a window size based on desired LOI size. We use 50x50 pixel windows for 25x25

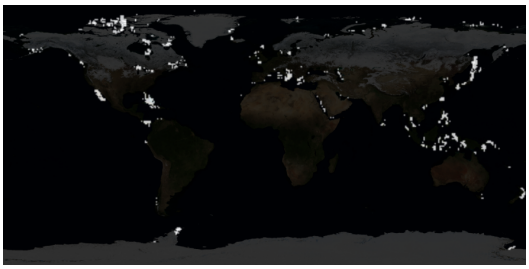

Fig. 3. The results of the ROI distinctness testing. The highlighted areas represent the top percentile of distinct pixels in the image.

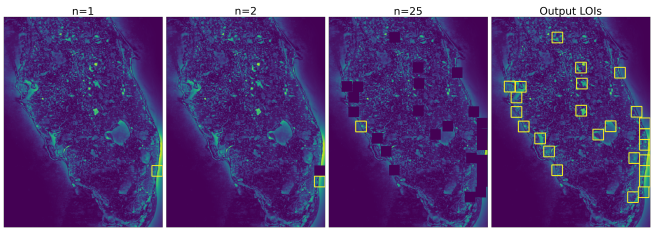

Fig. 4. The process for choosing LOIs within a region. The first image shows the highest saliency location in the image. The second image demonstrates the zeroing of the first location and the second highest saliency location in the image. The third image shows the 25th iteration of this approach. The final image shows the output boxes.

kilometer LOIs. The window is slid over the saliency map and the location with the highest sum of all pixels in the window is recorded as a LOI. The pixels in the window are then set to 0, reducing overlap, and the process is repeated to find the next LOI. This process is shown in Fig. 4. We generate 100 LOIs per region using this method.

Each LOI Detection (LD) network was trained using 5 random 200x200 km crops from every day in 2020 from Landsat images over the corresponding ROI, downsampled to a spatial resolution of 200 m. The dimensions were selected as representative of the sectioning of a higher-resolution 2592x1944 pixel image into smaller 1024x1024 pixel images for processing on orbit. The images are labeled using georeferenced images and recording bounding boxes and classes for every LOI in the image.

B. Region of Interest Classification

Using our assembled datasets, we built ROI and LOI networks that provide inputs to our OD filter. The first step in the image pipeline is classifying each image into a class referencing a predesignated ROI on the Earth, or an "uninteresting" class. We perform coarse ROI classification because many regions of Earth are uninteresting and the ROI classifier filters these without invoking LD, saving processing time and energy. The ROI classifier is a modified ResNet-50 convolutional neural network (CNN) [23] pre-trained on ImageNet and fine-tuned using our dataset.

C. Landmark Detection and Geolocation

After classifying the satellite image into a region, the next step is to identify geographic LOIs in the image and their absolute positions on the surface of the Earth. We accomplish this by using a separate LD network for each ROI with

classes specific to that ROI. The LD network locates LOIs in the satellite image and assigns classes to them, along with a confidence score. Each LOI class is tied to a latitude and longitude. The center point of the classified box found by the LD network is then mapped to the latitude and longitude of the class. The latitude, longitude, and score for each point are passed to the OD filter. The method used for LOI detection is a YOLOv8 CNN [24] pre-trained on the COCO dataset and fine-tuned using our dataset.

III. Orbit Determination

The OD filter takes in the information generated by the Image Pipeline in the form of unit vectors pointing from the camera to known landmark locations on the Earth's surface and outputs an estimated state comprising of the satellite's position and velocity. The filter consists of two main stages: a least-squares estimator and an Extended Kalman Filter (EKF).

The batch nonlinear least-squares algorithm described in [25] is used to determine an initial coarse state estimate from the first few landmark vectors produced by the Image Pipeline. This initial coarse estimate is used to provide a sufficiently accurate guess to initialize the EKF, which is then used to update the state estimate for all future measurements.

The initial state estimate from the least-squares algorithm is then used to initialize a square-root EKF [26], [27].

IV. Results

We evaluated our implementation of visual OD to show that we are able to rapidly localize a satellite to a precision on the order of 100 m in one pass over a ROI. We first describe the capability of ROI classification to accurately classify each image into the correct ROI or as "uninteresting." We then demonstrate the ability of the LD networks to find LOIs in an ROI image with an acceptable degree of error. Additionally, we characterize the latency of the LD network on representative computing hardware for nanosatellites. The performance of the image pipeline is then used to inform testing of the OD filter, and end-to-end simulation results of the entire pipeline are presented.

A. Region of Interest Classification

The ROI classification network achieves an average classification accuracy over 90% on the validation set. In the future, a more in-depth test set will be gathered and the ROI classification network will be refined.

B. Landmark Detection Performance and Latency

Fig. 5 shows the performance of a single trained network on the validation set. The network achieves a median LOI distance error of 325.6 m across all confidence scores, which is measured by taking the center point of the predicted location and comparing to the center point of the true location. Given that each pixel is 200x200 m, a median distance error of 325.6 m from the center of the LOI is less than two pixels of error. The detections corresponding to confidence scores greater than 0.96 made up more than

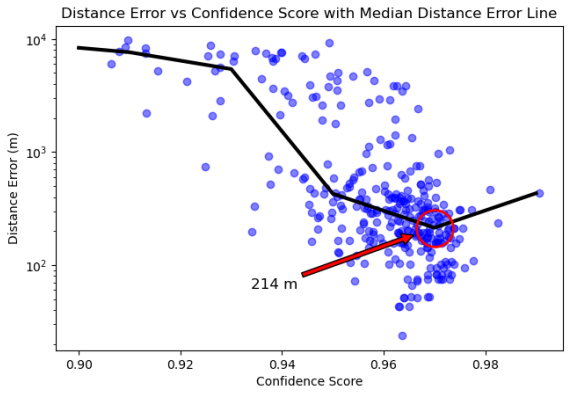

Fig. 5. The scatter plot shows the distance error for each location detected by the LD network and the associated confidence score. The plotted line is the median of 0.02 confidence-interval buckets, e.g. the circled point is the median of all points in the 0.96 to 0.98 confidence score range. The higher confidence values, i.e. the ones that will weigh more heavily in the OD filter, have a median distance error of 214 m, which is near the practical limit of what can be expected given imagery with a GSD of 200 m.

half of the total detections. These detections had a median distance error of 214 m, which is roughly a pixel of error.

The latency of LD model inference for a 1024x1024 pixel image on some representative devices is shown in Table II. The Jetson Orin running in 15W mode is representative of a larger CubeSat mission, and the Raspberry Pi Zero 2 is representative of a smaller CubeSat mission.

TABLE II
Model Latencies on Representative Devices

	Jetson Orin@15W	Raspberry Pi Zero 2
YOLOv8n-fp32	43.5 ms	3,527 ms
YOLOv8s-fp32	82.9 ms	24,814 ms
YOLOv8m-fp32	178.9 ms	86,262 ms
YOLOv8l-fp32	278.6 ms	OOM
YOLOv8x-fp32	475.5 ms	OOM
YOLOv8n-int8	17 ms	2,786 ms
YOLOv8s-int8	25.4 ms	6,077 ms
YOLOv8m-int8	55.6 ms	12,360 ms
YOLOv8l-int8	82.0 ms	24,100 ms
YOLOv8x-int8	135.3 ms	94,240 ms

C. Orbit Determination

The batch least-squares method can provide state estimation with two pairs of measurements separated by a few seconds. We used a dataset generated with satellite data that was extracted from an emulated ISS orbit using the Skyfield API [28] flying over Italy and Florida. Fig. 6 shows the error in measurement $\Delta y = y_{true} - y_{model}$ varying linearly with the resolution of the on-board camera.

D. Extended Kalman Filter performance

When fed with an initialized first state from least squares, the EKF can track the satellite's orbit to a precision on the order of hundreds of meters over the course of ten minutes. Velocities are tracked to an accuracy of the order of $0.1m/s$. Fig. 7 shows the norm of the position error at the last time

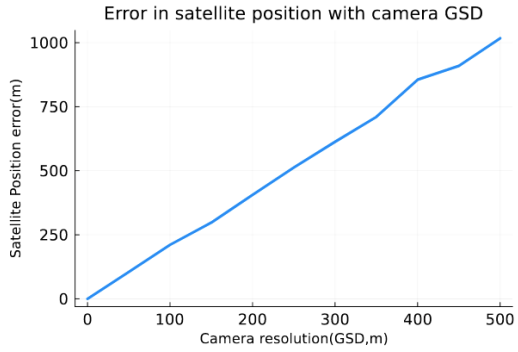

Fig. 6. Error of least squares optimization initial estimation increases almost linearly with the resolution of the on board camera

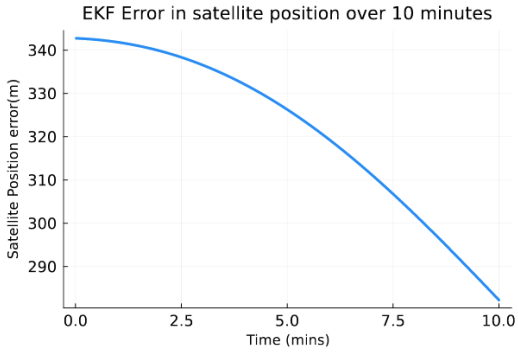

Fig. 7. EKF further improves estimates after least squares optimization from 350m to 290m after 10 minutes of continuous measurement.

step after running the EKF for 10 minutes with constant measurement acquisition at 1Hz.

V. Conclusions And Future Work

We demonstrated visual OD for a low-cost nanosatellite equipped with a low-cost camera and only onboard ML inference. The technique enables future satellites to quickly localize with no costly components or human-in-the-loop sensing. Building on this work, we will incorporate computation and energy constraints into the process and determine the impacts of these constraints on state error. Additionally, the networks will be further developed to be more robust to varying satellite altitudes, look angles, and camera characteristics. We also plan to use these same images for attitude determination using a combination of optical flow and LOI matching, along with inertial measurements. We plan to launch a demonstration of the technique in a PocketQube [11] containing an OV5640-C camera and microcontroller in Fall 2023. We hope to test the proposed method on orbit and measure its performance.

References

[1] J. Shao, L. Tang, M. Liu, G. Shao, L. Sun, and Q. Qiu, "BDD-Net: A General Protocol for Mapping Buildings Damaged by a Wide Range of Disasters Based on Satellite Imagery," *Remote Sensing*, vol. 12, no. 10, p. 1670, Jan. 2020.

[2] J. You, X. Li, M. Low, D. Lobell, and S. Ermon, "Deep gaussian process for crop yield prediction based on remote sensing data," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, Feb. 2017.

[3] R. Naushad, T. Kaur, and E. Ghaderpour, "Deep transfer learning for land use and land cover classification: A comparative study," *Sensors*, vol. 21, no. 23, 2021.

[4] "OEM719." [Online]. Available: https://novatel.com/products/receivers/gnss-gps-receiver-boards/oem719

[5] Y. Yang, X. Yue, and A. G. Dempster, "GPS-based onboard real-time orbit determination for leo satellites using consider Kalman filter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 2, pp. 769–777, Apr. 2016, conference Name: IEEE Transactions on Aerospace and Electronic Systems.

[6] "Space environment statistics." [Online]. Available: https://sdup.esoc.esa.int/discosweb/statistics/

[7] J. van den IJssel, J. Encarnação, E. Doornbos, and P. Visser, "Precise science orbits for the Swarm satellite constellation," *Advances in Space Research*, vol. 56, no. 6, pp. 1042–1055, 2015.

[8] M. A. Skinner, M. Coletti, M. C. Voss, T. Svitek, J. C. Lee, K. Auman, H. Patel, and E. J. Moyer, "Mitigating CubeSat confusion: Results of in-flight technical demonstrations of candidate tracking and identification technologies," *Journal of Space Safety Engineering*, vol. 9, no. 3, pp. 403–409, 2022.

[9] Z. Kang, B. Tapley, S. Bettadpur, J. Ries, P. Nagel, and R. Pastor, "Precise orbit determination for the grace mission using only gps data," *Journal of Geodesy*, vol. 80, pp. 322–331, 2006.

[10] B. Denby, E. Ruppel, V. Singh, S. Che, C. Taylor, F. Zaidi, S. Kumar, Z. Manchester, and B. Lucia, "Tartan artibeus: A batteryless, computational satellite research platform," 2022.

[11] S. Radu, M. S. Uludag, S. Speretta, J. Bouwmeester, A. Menicucci, A. Cervone, A. Dunn, and T. Walkinshaw, "PocketQube Standard," 2018.

[12] S. Caldwell, "12.0 Identification and Tracking Systems," http://www.nasa.gov/smallsat-institute/sst-soa/identification-and-tracking-systems, Oct. 2021.

[13] "5MP_Mega_M12_SPI_B0436," https://www.arducam.com/product/mega-5mp-color-rolling-shutter-camera-module-with-m12-lens-for-any-microcontroller/.

[14] "TEVI-OV5640," https://www.technexion.com/products/embedded-vision/camera-sensors/tevi-ov5640/.

[15] B. Denby and B. Lucia, "Orbital edge computing: Nanosatellite constellations as a new class of computer system," in *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, ser. ASPLOS '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 939–954. [Online]. Available: https://doi.org/10.1145/3373376.3378473

[16] "Data — Landsat Science," https://landsat.gsfc.nasa.gov/data/, Nov. 2021.

[17] "MODIS Web," https://modis.gsfc.nasa.gov/data/.

[18] N. Gorelick, M. Hancher, M. Dixon, S. Ilyushchenko, D. Thau, and R. Moore, "Google Earth Engine: Planetary-scale geospatial analysis for everyone," *Remote Sensing of Environment*, 2017.

[19] "December, Blue Marble Next Generation," https://visibleearth.nasa.gov/images/74218/december-blue-marble-next-generation.

[20] "NGA Geomatics - MGRS 100K Shapefile Downloads," https://earth-info.nga.mil/index.php?dir=coordsys&action=mgrs-100km-polyline-dloads.

[21] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[22] S. Montabone and A. Soto, "Human detection using a mobile platform and novel features derived from a visual saliency mechanism," *Image and Vision Computing*, vol. 28, no. 3, pp. 391–402, 2010.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[24] G. Jocher, A. Chaurasia, and J. Qiu, "YOLO by ultralytics," Jan. 2023.

[25] M. Holliday, K. Tracy, Z. Manchester, and A. Nguyen, "The v-r3x mission: Towards autonomous networking and navigation for cubesat swarms."

[26] K. Tracy, "A square-root kalman filter using only qr decompositions," 2022.

[27] S. Thrun, "Probabilistic robotics," *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, 2002.

[28] B. Rhodes, "Skyfield: High precision research-grade positions for planets and earth satellites generator," Astrophysics Source Code Library, record ascl:1907.024, p. ascl:1907.024, July 2019.