2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

VINSat: Solving the Lost-in-Space Problem with Visual-Inertial Navigation

Kyle McCleary[1], Swaminathan Gurumurthy[1], Paulo R.M. Fisch[1], Saral Tayal[1], Zachary Manchester[1], Brandon Lucia[1]

***Abstract*— Rapid growth in the number of nanosatellite deployments has heightened the need for rapid, cost-effective, and accurate orbit determination (OD). This paper introduces a solution to this "lost-in-space" problem that we call Visual-Inertial Navigation for Satellites (VINSat). VINSat performs OD using data from an inertial measurement unit (IMU) and a low-cost RGB camera. Machine learning techniques are used to identify known landmarks in images captured by the spacecraft. These landmark locations are then combined with IMU data and a dynamics model in a batch nonlinear least-squares state estimator to determine the full state of the spacecraft. We validate VINSat in simulation using real nadir-pointing imagery and find that 85% of simulated satellites are localized to under 5 km within 6 hours (4 orbits). This performance substantially surpasses that of ground radar, demonstrating significantly faster and more precise localization without any reliance on ground infrastructure.**

I. Introduction

Nanosatellites, including CubeSats, have been deployed in rapidly increasing numbers in recent years [1]. They are compact, cost-effective satellites that have opened up new possibilities for space research, technology testing, and educational initiatives. Their modular design and affordability have democratized access to space, allowing a wider range of organizations and individuals to participate in space exploration. They have found applications in diverse areas such as agriculture [2], land use classification, and disaster response [3].

Current methods for orbit determination (OD) of nanosatellites, such as Global Positioning System (GPS) receivers, radio ranging, and ground-based radar are large, expensive, imprecise, or time consuming. These limitations are underscored by data from the European Space Agency (ESA) that show a considerable fraction of CubeSats remain unidentified more than 250 days after launch [4].

Nanosatellites would greatly benefit from a solution to the "lost-in-space" problem, in which a spacecraft needs to determine its orbit relying solely on onboard computing and sensing with no prior state knowledge. This paper presents Visual-Inertial Navigation for Satellites (VINSat), a comprehensive solution to this problem that can determine a satellite's pose with kilometer-level accuracy within a few hours using only a low-cost camera, Inertial Measurement Unit (IMU), and on-board processing. For this paper, we will focus on a nadir-pointing satellite with an assumption that attitude is known and controlled.

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

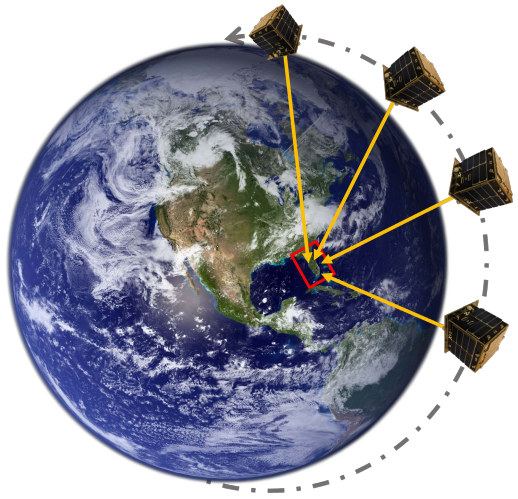


Fig. 1. VINSat solves the "lost-in-space" problem by determining a satellite's orbital parameters. It combines low spatial resolution image sensing and machine-learning inference with batch least-squares estimation and an accurate dynamics model to achieve kilometer-level position accuracy.

VINSat is a fast, cost-effective method that is compatible with the size, weight, and power limitations of nanosatellite hardware. VINSat addresses the challenge of OD by employing a computer vision model to identify landmarks on the Earth's surface and incorporating this information into a batch least-squares problem [5], [6], aligning the detected landmarks with their known 3D positions while accounting for the orbital dynamics between consecutive time steps. Our main contributions include:

- A complete end-to-end visual-inertial navigation pipeline for OD onboard nadir-pointing small satellites using only low-cost camera sensors and onboard computation.
- A large dataset of real and synthetic Earth imagery collected from satellites for training vision-based navigation algorithms for space applications.
- A thorough evaluation based on Monte Carlo simulation of more than 500 low Earth orbit (LEO) deployments demonstrating that VINSat localizes 85% of satellites to within 5 km in just 6 hours; substantially faster and more precise than using ground-based radar.

This paper proceeds as follows: Section II discusses previous work on OD. Section III describes each subsystem making up the end-to-end system in detail. Section IV then presents an evaluation framework and the results obtained

with the system. Finally, Section V summarizes our conclusions and directions for future work.

II. Related Work

In this section, we provide a brief overview of the existing approaches to OD.

A. GPS Receivers

GPS receivers are the standard for precise OD [7], [8], [9], but are ill-suited for nanosatellites, such as CubeSats or PocketQubes [10], [11], due to their size and cost. A space-rated version of the NovAtel OEM719 GPS receiver [12] is commonly used in CubeSats due to its relatively small size, low cost, and ability to operate at the extreme speeds (roughly 7.5 km/s) of a satellite in LEO. These receivers cost around $5000. Modules that use these receivers to perform OD on-orbit cost even more. One such system, [13], uses a NovAtel OEM719 for position data to perform OD and costs $10k-$19k. Standard GPS receivers for use on Earth are restricted to speeds less than 515 m/s and altitudes less than 18 km due to limits placed by the Coordinating Committee for Multilateral Export Controls (CoCom), its successor the Wassenaar Arrangement, and the Missile Technology Control Regime (MTCR) [14].

B. Ground Radar

An alternative to GPS is radio ranging from ground stations or radar [8], [15], which can take weeks to months to determine a satellites orbit after launch, and may even fail to identify a satellite in a cluster [15]. Radio ranging from ground stations requires significant infrastructure, and positions from radar typically have errors of 20 km [16]. Table I compares the existing commonly used methods to our approach.

TABLE I
Comparison of Orbit Determination (OD) Methods

Property/Method	GPS OD	Ground Radar	Visual OD
Largest Dimension	96 mm [13]	Off Satellite	< 50 mm
Mass	109 g [13]	Off Satellite	< 15 g
Power	1-2 W [12]	Off Satellite	< 5 W
Cost	$\sim$ $10 k [13]	$0 - 10 k	$< $100
OD Time	Secs [17]	Wks-Mos [18]	Secs-Hrs
Precision	1.5 m [17]	$\sim$ 10 km [16]	$\sim$ 1 km

C. Visual Methods

Prior work on visual satellite navigation such as [19] and [20], obtains coarse satellite position estimates using only visual inputs from cameras and classical computer vision keypoint-matching techniques, such as SIFT [21], FLANN [22] and RANSAC [23]. These approaches have several limitations, including lack of robustness to clouds, the use of pre-extracted coastlines for localization, and localization errors of a few degrees of latitude and longitude.

III. System Design

This section describes the architecture of the VINSat system. We provide a high-level overview, followed by a detailed description of each subsystem.

A. System Overview

The VINSat system, as depicted in Fig. 2, has two main components: an image-processing subsystem responsible for extracting Earth landmarks from captured imagery and a batch least-squares optimization solver that calculates the satellite's orbit based on these landmarks.

Initially, images are captured by a camera at a resolution of 4608×2592 pixels. Identifying landmark correspondences in the captured images is challenging, particularly without prior knowledge of the satellite's pose, and is further compounded by the computational constraints imposed by a nanosatellite's limited power and processing capacity.

VINSat geolocates captured imagery by matching image features to landmarks on Earth. First, the system identifies the coarse geographic region over which the satellite is orbiting. Second, it matches ground landmarks with known locations to pixels in the image. VINSat performs region identification using a Region Classification (RC) Network. An RC network is a deep neural network that is trained to process each captured image at a downscaled resolution of 640×360 pixels to identify the region of Earth that the image depicts. Additionally, the RC network eliminates uninteresting images, such as those containing only clouds or ocean. These outputs are subsequently used to route the full-resolution images to specialized landmark detection (LD) deep neural networks, each trained for a specific region. The LD networks produce a mapping from pixel coordinates to landmarks with known location coordinates on Earth. These coordinates are the input to a batch optimization solver that produces an estimate of the satellite's orbit. Further details of each subsystem are elaborated upon in the sections that follow.

B. Region Classification

The RC network takes an image as input and produces the regions that the image most likely depicts as output. The purpose of the RC network is to narrow the scope of the later search for landmarks to a small set of regions on Earth, rather than all of Earth.

The RC network's output classes are region identifiers corresponding to regions defined in the Military Grid Reference System (MGRS), a NATO international standard for locating points on the Earth. The largest regions of the MGRS divide the Earth into a grid delineated by 22 North-South regions and 60 East-West regions. Each region is typically six degrees of longitude by eight degrees of latitude, with some variation near the poles.

The RC network does not consider all regions in the MGRS, instead focusing on the most *salient* 16 regions. VINSat computes the saliency of a region by computing the cross-correlation of the NASA Blue Marble imagery data available for that region. Fig. 3 shows a map of the

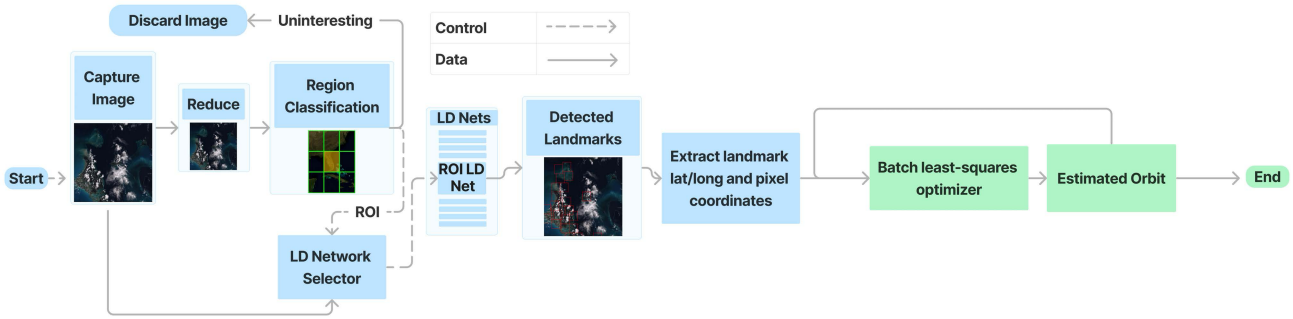


Fig. 2. End-to-end block diagram of VINSat pipeline. The vision block is highlighted in blue and the state estimation block is highlighted in green. Captured images are first classified into regions of interest (ROIs). These ROIs are used to feed the captured image to the appropriate landmark detection (LD) networks. The identified landmarks, their positions in the image, and their associated confidences are passed to the batch least-squares optimizer. The batch least-squares optimizer uses the information to estimate the orbit of the satellite.

Fig. 3. Saliency map from which regions of interest were selected. Brighter areas exhibit high saliency. These areas were used to select regions of interest for the region classification and landmark detection networks.

computed saliency of points on Earth. Based on a Monte Carlo simulation of 1000 typical random nadir-pointing LEO satellites, the average time for a CubeSat to pass over one of these regions is about 63 minutes.

The RC network is based on the EfficientNet-B0 architecture, a highly efficient neural-network model suitable for online inference on each captured image. To adapt it for our setting, we replace its output layer with a sigmoid activation function for region detection.

C. Landmark Detection

Each LD network takes an image as input and produces a set of landmarks contained within the image and their pixel locations as output. VINSat selects landmarks based on saliency of features within a region of interest (ROI). We use OpenCV's saliency API on NASA Blue Marble image data for each of our 16 regions of interest. We sequentially select the top 500 most salient boxes in each region of varying sizes between 5 km and 50 km. An example of this process is shown for boxes of 25×25 km in Fig. 4.

VINSat's LD network is based on the YOLOv8s object detection model, trained to produce bounding boxes around landmarks within an image. We convert landmarks and detection bounding boxes to points (which is needed for OD) by using their center point.

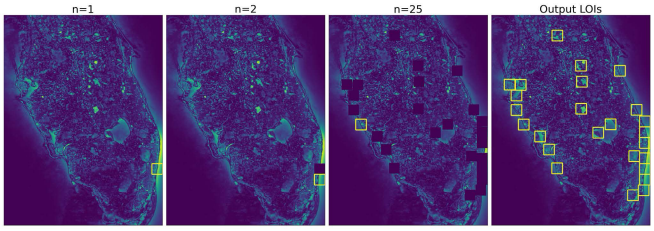


Fig. 4. Landmarks from the saliency map of an individual region. Each yellow rectangle is a 25×25 km bounding box of a salient landmark. The saliency-based landmark-selection process determines the landmark detection network's classes for each region. 500 landmarks are selected for each region of interest.

D. Orbit Determination

The primary objective of the OD pipeline is to use measurements from the image processing subsystem to generate accurate estimates of the satellite's state, which includes its position $\mathbf{r}$, and velocity $\mathbf{v}$. The estimation problem is formulated as a batch least-squares optimization problem [5], [6], aiming to minimize the residuals arising from satellite dynamics across consecutive time steps and the discrepancies between projected and observed landmarks on the camera frame.

The inputs to the system are the 3D locations of the detected landmarks in Earth-centered inertial (ECI) coordinates $\mathbf{p} = (x, y, z)$ and the pixel position of the corresponding detections in the image are $\mathbf{w} = (u, v)$.

The dynamics error term, ψ_t, is a summation of two key components: squared errors on the predicted and actual position and velocity. The predicted position ($\hat{\mathbf{r}}$) and velocity ($\hat{\mathbf{v}}$) are obtained by rolling out the orbital dynamics from the previous state.

$$\psi_t = \|\hat{\mathbf{r}}(\mathbf{r}_{t-1},\mathbf{v}_{t-1}) - \mathbf{r}_t\|^2_{Q_r} + \|\hat{\mathbf{v}}(\mathbf{r}_{t-1},\mathbf{v}_{t-1}) - \mathbf{v}_t\|^2_{Q_v}, \tag{1}$$

where Q_r and Q_v are weight matrices corresponding.

The second major component in the batch optimization problem is the camera projection error term, ϕ_m. Given the

satellite's estimated position $\mathbf{r}$ and assuming nadir orientation $\mathbf{q}_n$, along with the camera matrix K, we can project 3D landmarks $\mathbf{p}$ to their expected pixel positions $\hat{\mathbf{w}}$ in the camera frame. The error term ϕ_m quantifies the discrepancy between these predicted pixel positions and the actually observed positions $\mathbf{w}$:

$$\phi_m = \|\hat{\mathbf{w}}(\mathbf{r_m}, K, \mathbf{p_m}) - \mathbf{w}_m\|_R^2, \tag{2}$$

where R is a weight matrix. One challenge with this error term is its sensitivity to outliers, particularly when the landmark measurements are noisy. Thus, instead of using a constant R, we compute it as proposed in [24] to obtain an adaptive kernel :

$$R_k = c^2 \left(\frac{(\epsilon_k^p / c)^2}{|\alpha - 2|} + 1 \right)^{1 - \frac{\alpha}{2}}, \tag{3}$$

where R_k is the k-th diagonal term in R, ϵ_k^p is the corresponding residual term, and α and c are hyperparameters that control the shape and scale of the kernel, respectively. We set c to the median of the residuals to automatically adapt the scale of the kernel to the specific problem. We compute $\alpha = \max(2 - 2 * i/5, -1)$, as a function of the optimizer iteration count, i. $\alpha = 2$ corresponds to a least-squares kernel, and lower $\alpha's$ correspond to increasingly more robust kernels. Setting $\alpha = \max(2 - 2i/5, -1)$ allows us to compute an initial estimate using all points and then progressively make the estimates more robust to outliers.

Algorithm 1 Batch Least-Squares Optimization $LSQ(\mathbf{r_m}, K, \mathbf{p_m}, \mathbf{r}_{t-1}, \mathbf{v}_{t-1}, \Delta t, \mathbf{r_t}, \mathbf{v_t})$

for i *in* num_iters **do**
  $\hat{\mathbf{w}}_m, J_{g,m} = CameraProjection(\mathbf{r_m}, K, \mathbf{p_m})$
  $\hat{\mathbf{r}}_t, J_{d,t}, H_t = Dynamics(\mathbf{r}_{t-1}, \mathbf{v}_{t-1})$
  Compute ψ_t, ϕ_m using Eq 1, 2.
  $\hat{\mathbf{v}}_:, \hat{\boldsymbol{r}}_: = LM\_Optimizer(J_{g,:}, J_{d,:}, \psi_:, \phi_:)$
end for
return $\hat{\mathbf{w}}, \hat{\boldsymbol{r}}$

Finally, we aggregate these costs across detections and time-steps and solve a joint optimization problem as follows:

$$\min_{(\mathbf{r},\mathbf{v})_{1:N}} \sum_m \phi_m + \lambda \sum_t \psi_t \tag{4}$$

We solve this optimization problem over all the poses jointly using a Levenberg-Marquardt (LM) method [25]. We provide an overview of the system for solving the batch least-squares problem in Algorithm 1.

IV. Evaluation

The purpose of this evaluation is to demonstrate the ability of the VINSat system to sufficiently perform OD from typical CubeSat orbits in a reasonable time frame. This section details evaluations of the individual components of VINSat, as well as the end-to-end system. Our simulation results show that VINSat achieves kilometer-level OD in just a few orbits.

To evaluate VINSat, we require a large dataset of images captured from a low-cost and wide field of view camera on a satellite orbiting Earth, as well as images containing clouds, which are typically discarded by satellite imagery distributors. Unfortunately, such a dataset is not currently available. Consequently, we divide our evaluation into three main components:

1) *Cross-Dataset Generalization*: We test generalization to different cameras and data sources by training on Sentinel 2 imagery and testing on Landsat imagery.
2) *Ablations and Sensitivity Analysis*: We perform ablation experiments to understand the importance of various design decisions on the batch optimizer.
3) *Simulation*: We simulate satellites in random polar and ISS-like orbits to evaluate the end-to-end pipeline.

A. Cross-Dataset Generalization

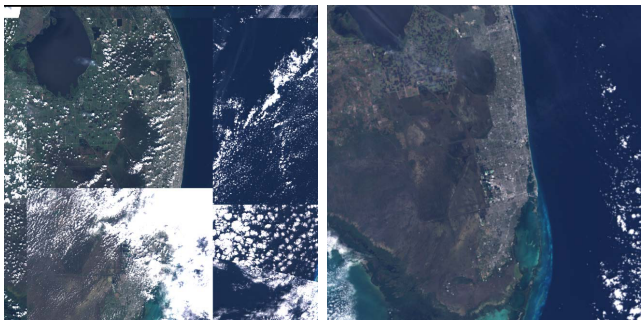
We test cross-dataset generalization by training on Sentinel 2 imagery and testing on Landsat imagery. We do this by first training on 2000 randomly mosaiced Sentinel-2 images for each of the 16 regions. The images are made from randomly selecting points in the region of interest, buffering around the point, mosaicing random Sentinel 2 images captured between 2020 and 2022, and exporting the image at a scale of 150 meters per pixel, similar to our reference camera. An example of one of these images is shown on the left of Fig. 5. We then validate the performance of each network on 500 Sentinel 2 images made the same way, but from data captured between 2018 and 2019. 250 raw Landsat 8 and 250 raw Landsat 9 scenes captured during 2023 are used for testing. An example Landsat 8 scene can be seen on the right of Fig. 5. All images were downloaded from Google Earth Engine [26] in GeoTIFF format containing affine transformations of pixel points to ground points to maintain accurate ground truth when labeling landmarks.

For clarity the training, validation, and test datasets **for each region** are listed again below:

- *Training*: 2000 Sentinel 2 mosaics made from imagery captured between 2020 and 2022.
- *Validation*: 500 Sentinel 2 mosaics made from imagery captured between 2018 and 2020.
- *Test*: 250 Landsat 8 and 250 Landsat 9 images captured during 2023.

a) Region Classification Network: The RC network dataset is annotated by projecting landmarks onto the camera's image plane and identifying regions with visible landmarks. Each image is assigned a 16-value multi-hot label corresponding to the regions containing visible landmarks in the image. The RC network is validated by measuring the performance of the network on the validation dataset using network precision, recall, and F1 score.

The RC network achieves an overall accuracy of 99.2% on the validation set. The mean precision for all classes is 0.95, the mean recall for all classes is 0.97, and the mean F1 score for all classes is 0.96. These values demonstrate that the appropriate regions have high likelihood of being recognized in an image and passed to the corresponding landmark detection networks.



(a) Mosaiced imagery from Sentinel satellite (b) Imagery from Landsat satellite

Fig. 5. Comparison of Sentinel and Landsat imagery. The many subtle differences between imagery sources, lighting conditions, and seasons lead to a challenging landmark detection problem.

b) Landmark Detection Networks: Each LD network dataset is annotated by projecting landmarks onto the camera's image plane and identifying the pixel locations of each landmark. We locate the center point and corners of each landmark in the image and draw a bounding box around it, ensuring that the center point of the label aligns with the center point of the landmark. Each LD network is evaluated by measuring the performance of the network on the test dataset by measuring mean pixel error of the center points of detected landmarks at varying confidence thresholds. Additionally, we measure the ratio of detections included at varying confidence thresholds.

We report results for the mean pixel error and ratio of points encompassed across all LD networks for confidence thresholds of 0.7, 0.8, and 0.9 in Table II. From the table it is clear that a confidence threshold of 0.9 is too high as it includes a very small sample of the detections while not improving mean pixel error. We choose 0.8 as a confidence threshold as the majority of detections are included, but the mean error is improved over a threshold of 0.7.

These results suggest that both networks exhibit robust generalization to substantial variations in clouds, seasons, and image characteristics.

TABLE II

LD NETWORKS MEAN PIXEL ERROR AND RATIO OF INCLUDED POINTS AT VARYING CONFIDENCE THRESHOLDS.

	0.7	0.8	0.9
Mean Error (pixels)	1.79	1.66	1.87
Ratio of Included Points	0.86	0.65	0.03

B. Ablations and Sensitivity Analysis

We perform ablation experiments to understand the importance of various design decisions.

1) Effect of the orbital dynamics costs: We conducted tests to evaluate the influence of orbital dynamics terms on the performance of the batch optimizer using the end-to-end pipeline validation dataset and the detections from the image pipeline. We observe that, in the absence of the dynamics terms, the resulting trajectories have an RMS position error of 6.11 ± 3.32 km as opposed to 1.60 ± 0.87 km with the dynamics terms. Our observations highlight the critical role played by the dynamics terms.

2) Effect of outlier rejection: While our RC and LD networks excel at identifying landmarks on the Earth's surface, the presence of outliers in the detections significantly hampers OD. We use two primary methods for outlier rejection: confidence thresholding to remove detections below a specific confidence threshold and an adaptive cost kernel.

Table III presents the outcomes of confidence thresholding for ten randomly sampled LEO orbits, indicating that optimal results are achieved around a threshold of 0.8. This threshold strikes a balance between having enough data points for convergence while effectively controlling the impact of outliers. Higher confidence thresholds result in poorer performance due to data scarcity, whereas lower thresholds lead to an excess of outliers that disrupt convergence.

We also test our method without the adaptive kernel, replacing it with a simple L2 loss instead. This results in an RMS position error of 2.39 ± 1.44 km, which is much higher than our original results. We observe that the outlier rejection done by the adaptive kernel is very effective and contributes to significant performance improvements.

TABLE III

RMS OD ERROR PERFORMANCE AND CONFIDENCE THRESHOLDS

RMS OD Error/Confidence	**0.3**	**0.6**	**0.8**	**0.85**
Average (km)	2.75	2.14	1.60	2.06
Median (km)	2.50	1.69	1.53	1.72
Std. Dev. (km)	1.71	1.18	0.87	1.30

We observe that the performance of the batch least-squares optimizer improves monotonically with increasing density of detections at the same noise level. We test this by downsampling detections from a three hour orbit by varying it from 10% of detections per frame to 80% of the total detections per frame with the same noise level. An increase in the measurement density leads to a natural decrease in error. However, as density continues to rise, the error asymptotically approaches zero, resulting in diminishing returns after 60% detections per frame. Table IV shows the effect of detection density for 10 randomly sampled LEO orbits.

TABLE IV

RMS OD ERROR PERFORMANCE AND DETECTION DENSITY

RMS OD Error/Detection Density	**0.1**	**0.2**	**0.4**	**0.8**
Average (km)	6.67	6.92	2.32	1.80
Median (km)	6.26	3.64	1.68	1.69
Std. Dev. (km)	4.93	8.31	1.28	0.65

3) Effect of detection signal noise: In Table V, we illustrate the impact of pixel error on OD error. To simulate error originating from the detection network, we introduce white Gaussian noise to the camera pixel values of ten randomly sampled LEO orbits. The noise's standard deviation ranges from one to eight pixels. During these experiments, the OD error spans from 0.2 km to 1.35 km.

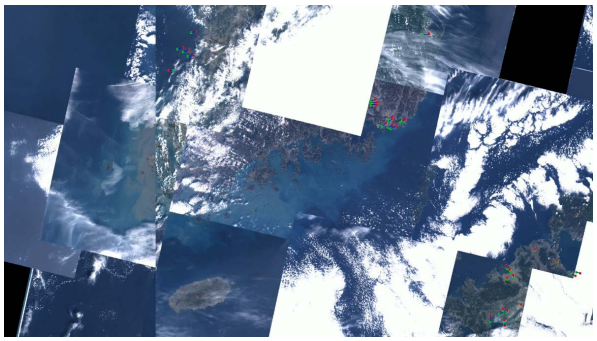


Fig. 6. View of a simulation image. Predicted landmarks are green, ground truth landmarks are red, and a blue line connects each associated predicted landmark and ground truth landmark. The landmark detection networks are capable of detecting landmarks with minimal error in non-cloud obscured portions of images.

TABLE V

RMS OD ERROR PERFORMANCE FOR DIFFERENT PIXEL WHITE NOISE STANDARD DEVIATION

RMS OD Error/Pixel noise σ	$\mathbf{1px}$	$\mathbf{2px}$	$\mathbf{4px}$	$\mathbf{8px}$
Average (km)	0.20	0.38	0.64	1.35
Median (km)	0.09	0.20	0.37	0.75
Std. Dev. (km)	0.17	0.32	0.49	1.11

C. Simulation

The simulation environment works by first generating a random near-polar or ISS-like orbit. This orbit is propagated at 1 Hz and the satellite pose is used at each timestep to determine the view of the Earth based on the reference camera intrinsics. Random Landsat 8 imagery is mosaiced to create an image in the view of the camera on the satellite. Images are captured at a rate of $\frac{1}{5}$ Hz. The image is then processed through the pipeline and detections are recorded. The detections are passed to the batch optimizer and the satellite's position is estimated.

We find that during simulation the LD networks sometimes struggle amidst significant cloud cover, but with the iterative elimination of "bad" classes (i.e. those with consistently high error) the LD networks consistently achieve detections with pixel error less than 10 pixels (1.5 km), often achieving low-single-digit pixel error. Occasional outliers with significant pixel error sometimes cause difficulty for the batch estimator. These outliers can be reduced by improving training of the LD networks and having more robust outlier detection.

An example view with detections and errors is shown in Fig. 6. Over 500 simulated random satellites were used to generate the cumulative distribution function-like plot shown in Fig. 7. The plot demonstrates that 85% of simulated nadir-pointing satellites reached a position error of less than 5 km in under four orbits. For satellites moving at 7.5 km/s, that corresponds to less than one second of position error.

V. Conclusions and Future Work

In this work, we introduce VINSat, an OD method that is, to the best of the authors' knowledge, the first fully autonomous, vision-based solution to the "lost-in-space" problem. Notably, our approach eliminates the need for expensive and bulky GPS receivers or time-consuming ground-based radar methods. We have also developed an evaluation pipeline and datasets using openly accessible tools, which we have released alongside this paper. We hope that this spurs further research and development in this important emerging field.

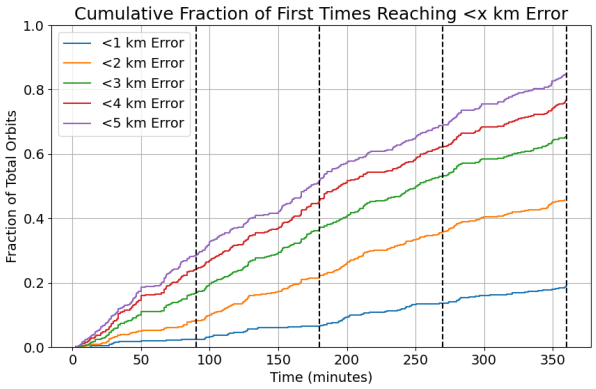


Fig. 7. Cumulative distribution of satellite position error vs. time. The y-axis corresponds to the fraction of total simulated satellites and the x-axis corresponds to the time of the simulation in minutes. Each separate line corresponds to a localization threshold between one and five km. The time stops at six hours, equivalent to roughly four full orbits around the Earth. The dashed vertical lines represent 90 minute intervals, or around 1 orbit. After over 500 simulations, 85% of satellites were localized with less than 5 km error.

While VINSat represents an important first step toward addressing the "lost-in-space" problem, we believe there exists significant scope for improvement and future work. The first, and possibly easiest, improvement is to simply increase the number of regions of interest. Furthermore, the batch least-squares problem only solves for the satellite position in the current setup. A natural next step would be to modify the problem to jointly solve for the position and the attitude of the satellite. Methodologically, both the LD and RC networks could benefit from architectural modifications that better utilize the sequential nature of the pipeline. In terms of evaluations, the end-to-end VINSat setup needs to be tested on real-world data with more refined sensor models that consider imperfections, such as image blur and IMU bias. Due to the lack of open datasets, this has been challenging. A natural next step from there would be to test the pipeline on hardware and finally on-orbit.

VI. Acknowledgments

This work was funded in part by NSF CPS Frontier Award #2111751 and a grant from the Sloan Foundation. Additionally, the authors would like to thank all those that helped in the course of this research, including CMU's Design-Build-Fly Laboratory course for their refinement of concepts in this paper.

References

[1] E. Kulu, "Nanosatellite launch forecasts-track record and latest prediction," 2022.

[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] 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.

[4] F. Letizia, "Results from esa's annual space environment report, july 2019, presented as a key-note address at the advanced maui optical and space surveillance technologies conference, held in wailea, maui, hawaii, september 2019. used by permission from esa," 2019.

[5] D. G. Robertson and J. H. Lee, "A least squares formulation for state estimation," *Journal of process control*, vol. 5, no. 4, pp. 291–299, 1995.

[6] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.

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

[13] "GNSS Receiver Module (GPSRM 1) Kit," http://www.pumpkinspace.com/store/.

[14] "MTCR Annex - MTCR," https://www.mtcr.info/en/mtcr-annex.

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

[16] "Planet Labs public orbital ephemerides," https://ephemerides.planet-labs.com/.

[17] 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.

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

[19] M. Straub and J. A. Christian, "Autonomous optical navigation for earth-observing satellites using coastline matching," in *AIAA Guidance, Navigation, and Control Conference*, 2015, p. 1334.

[20] L. M. Shockley and R. A. Bettinger, "Real-time aerospace vehicle position estimation using terrestrial illumination matching," in *2021 IEEE 8th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, 2021, pp. 505–509.

[21] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2. Ieee, 1999, pp. 1150–1157.

[22] M. Muja and D. G. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 11, pp. 2227–2240, 2014.

[23] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[24] J. T. Barron, "A general and adaptive robust loss function," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4331–4339.

[25] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly of applied mathematics*, vol. 2, no. 2, pp. 164–168, 1944.

[26] 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.