Quadchart

19/02/24

Demonstrating Visual-Inertial A&OD & On-Orbit Edge Computing

Progress summary

80 days before May 1st

Updates

- Vision:
  - Dataset download from 23 most salient regions
  - Landmark pruning for to identify ideal landmark size
  - Started hyperparameter tuning for LD
  - Looking into custom loss function focusing on pixel error
- Estimation integrated testing:
  - Generated camera vectors using landmark and satellite ground-truth through vector transformations
  - Tested batch optimizer to validate attitude estimation
  - Created small test setup to validate pixel to camera vector transformation
- Validation:
  - SIL environment setup between PyCubed and Simulation
  - First SIL test for detumbling control
  - Gyroscope noise analysis

Blockers

- Computing resources for LD training
  - ECE Community Compute Clusters
  - Pittsburgh Supercomputing Center
  - ROBO Cluster

Weekly Plan

- Vision
  -
- Estimation
  - Continue development on batch optimiser
- FSW development
  - Work on PyCubed-Jetson communication
  - Finished PyCubed tasks for Alpha version

Interface dependencies

- Final CAD of the CubeSat for test bed design

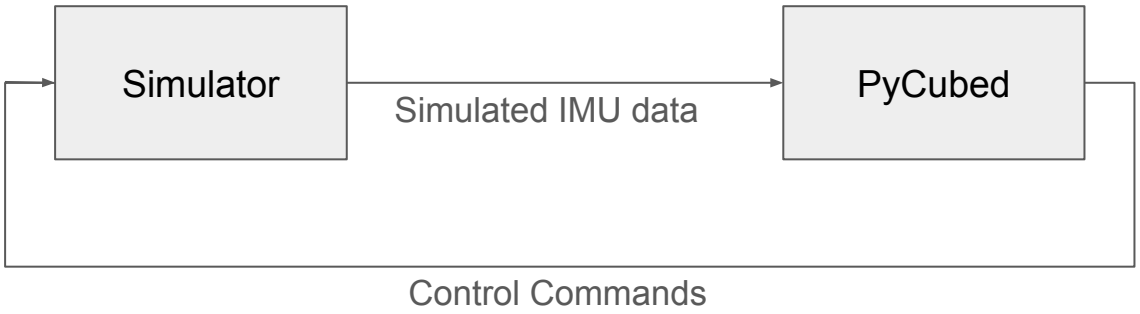
Vision

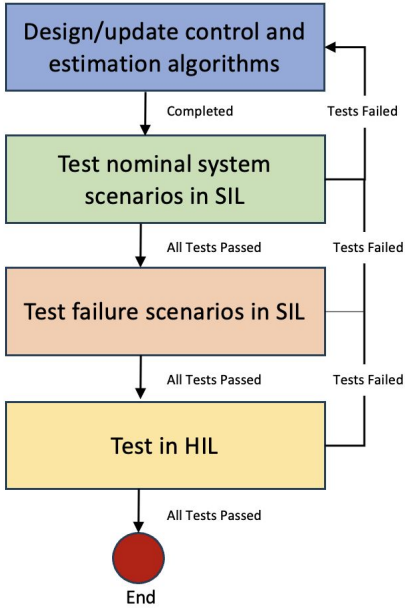
- **Dataset**
  - Downloading data from 23 regions -> Dataset Download Report
    - 2000 sentinel 2 train images + 500 sentinel 2 val images + 250 landsat 8/9 test images
    - Created tutorial and assigned download tasks for everyone in payload-GNC team
  - Landmark Pruning: YOLO dataset with 5000 annotated landmarks of various size and saliency
- **Training**
  - **RC:**
    - Trained multi-label classifier for 3 classes. Achieved ~85% accuracy trained on sentinel images, tested on landsat with GSD of 328. 565x565 sized images
    - Adding more classes to check performance of network - attempting to add 3 more classes this week
  - **LD:**
    - Set up hyperparameter tuning scripts for non-augmentation parameters
    - Iteratively training for X epochs then pruning undetectable landmarks and resuming training
    - Ran experiments to compare results on different input image resolution
    - Looking into custom loss function that focus on pixel error

Integrated testing

- Created small setup to test camera vector generation using image coordinates
- Verified transformation equations using actual x,y and depth information to find corresponding pixel coordinates and vice versa.
- Tested batch optimisation using groundtruth satellite ECEF and landmarks detected.
- Process
    - Get landmark lat long and convert to ECEF, get satellite groundtruth ECEF
    - Subtract the two vectors to get vector pointing from Landmark to satellite, invert to get vector from satellite to landmark
    - Convert this vector into camera frame to get camera vectors
    - Use ECI coordinates and these vectors as inputs to estimator

GNC Software Validation



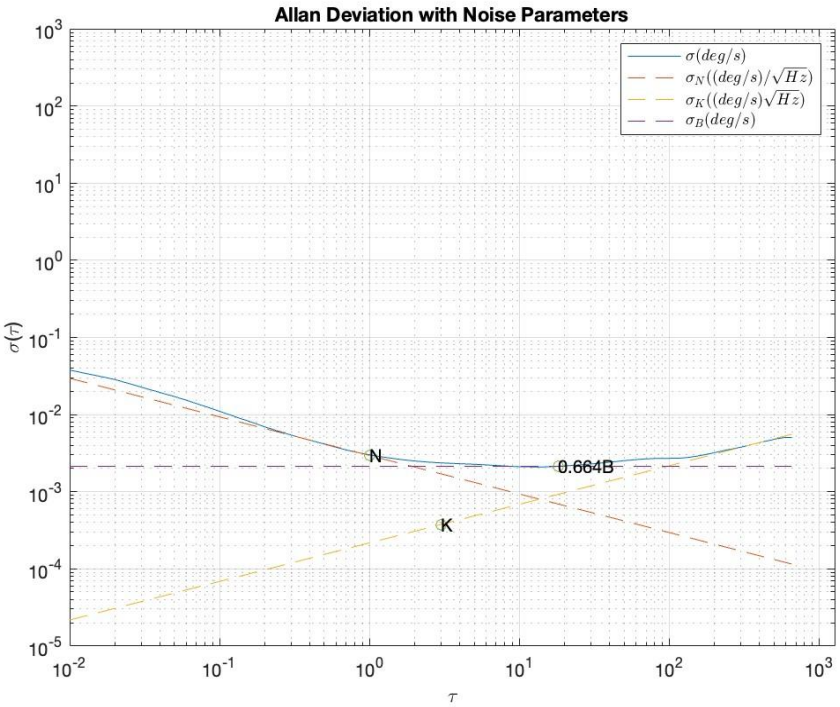


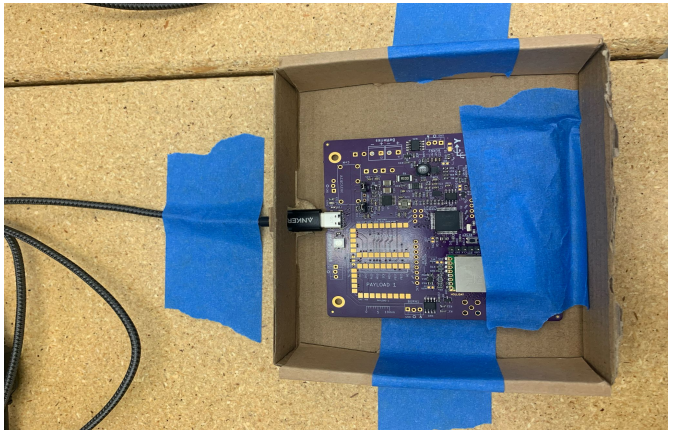
Software in Loop environment setup completed.

- Sensor readings generated in real-time on Simulator
- Readings are sent to PyCubed with Magnetic Control software running via Serial Communication
- Magnetic Control software processes the sensor readings and generates the control commands
- Simulator computes the next state for the satellite using the received control commands

Preliminary testing for Detumbling Control completed.

Gyroscope Noise Analysis





6 hours of datalogging of stationary IMU

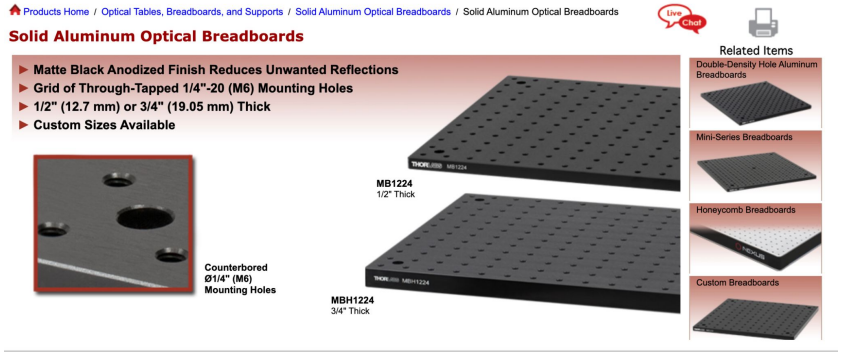
- Allan variance for gyroscope noise parameter analysis, Parameters: N (angle random walk), K (rate random walk), B (bias instability)
- Helps us identify noise sources in stationary gyroscope data clusters
- Will be used for modelling the gyroscope accurately

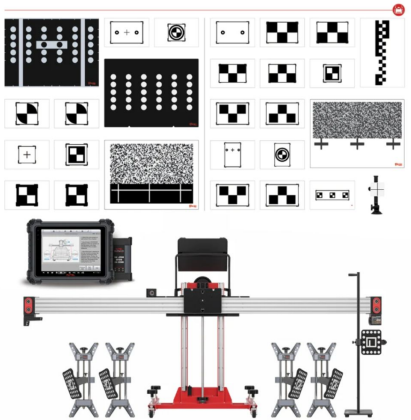
Calibration

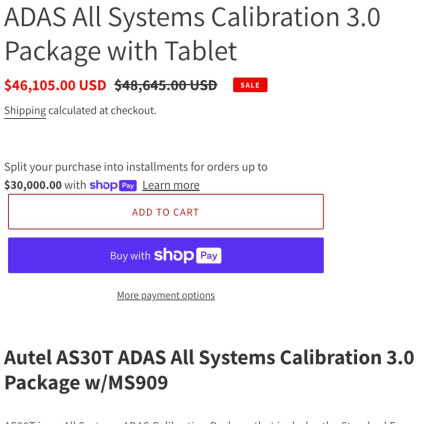
Equipment needed:

- Optical table / Optical Breadboard
- Precision rotation mounts
- Precision camera mount
  - https://www.bhphotovideo.com/c/product/5546-REG/Manfrotto_131TC_131TC_Tablemount_Geared_Column.html/?ap=y&ap=y&smp=y&smp=y&lsft=BI%3A514&gad_source=4&gclid=CjwKCAiA8sauBhB3EiwAruTRJrSTaedHKsTycGV0FSDZEnZGpjtnRIsGlNIlbAUzT8mNo6MbhC6JYRoCH4YQAvD_BwE

- Machined template for camera







12/02/24

Demonstrating Visual-Inertial A&OD & On-Orbit Edge Computing

Progress summary

80 days before May 1st

Updates

- Vision:
  - New datasets
  - YOLO training result
  - LD net inference updates
- Estimation integrated testing:
  - Testing and validating coordinate transformations and batch optimiser modules with Landsat images
- Testbed:
  - Preliminary research on V2 darkroom design
  - Research about multi camera calibration and color/brightness correction

Blockers

- Camera finalization (avionics?)
- Differences between training pictures and test photos → need to test with the new camera

Weekly Plan

- Vision
  - Develop YOLO training plan with region/parameter priority
  - Get high-performing (generalizable) LD model for one region for orbital pass test in V2
  - Complete integration of RC + LD inference on Jetson
  - Improve RCnet accuracy with new dataset
- Estimation
  - Test the coordinate transformations using real world points seen from the reference camera to verify the camera unit vectors
  - Test the batch optimizer with simulated world coordinates and camera matrices.
- MCM FSW testing
  - Test and validate Control software on PyCubed with simulated sensor readings

Interface dependencies

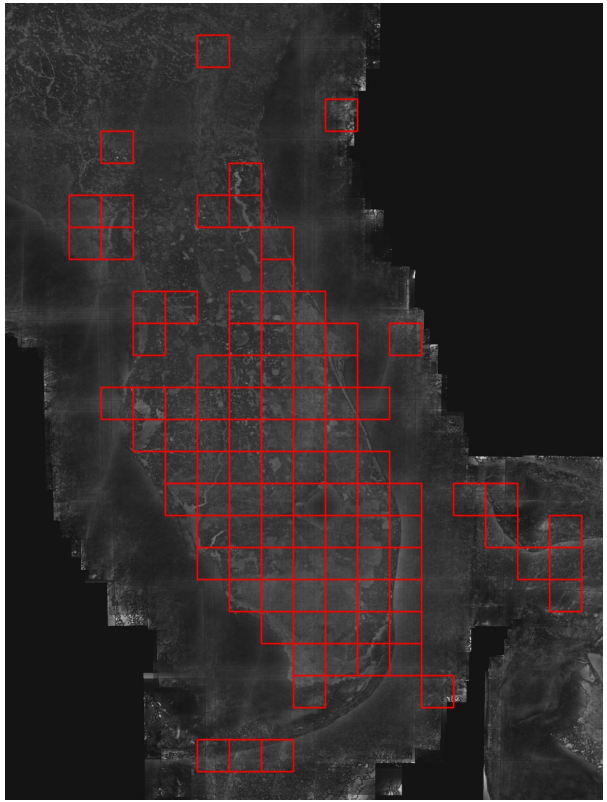
-

Vision

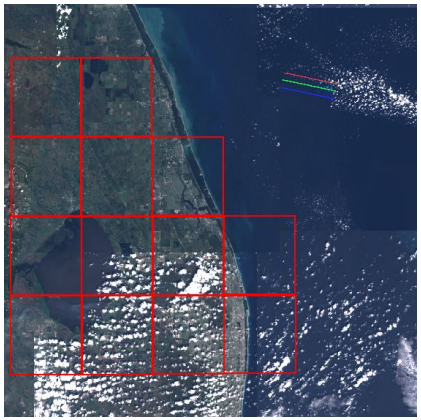
- **Dataset**
  - Integrated new downloader to download sentinel 2 mosaics to current pipeline
  - Generated new YOLO datasets
    - 2000 sentinel 2 train images + 500 rotated landsat images
    - Different landmark sizes: 25x25km, 30x30km, 35x35km
- **Training**
  - **RC:**
    - Implemented inference pipeline to work on jetson
    - Trained Efficientnet-b0 on old data but hit ceiling at ~40% mAP. Training with new data to start today - experimenting to increase mAP to ~70% this week
  - **LD:**
    - Trained YOLO models of different sizes (S, M, L) on Sentinel train/Landsat val set
      - Diverse augmentation params needed to generalize
      - mAP-50 above **0.9**, mAP50-95 above **0.77** for all three model sizes after 300 epochs
    - Evaluation script to track MSE, missed detections, extraneous detections across val set
    - Next step: prune landmarks and tune landmark size and number based on centroid pixel error
- **Jetson**
  - LD inference class updated to provide diverse output, testing script updated to provide evaluation of mean square error (MSE) for centroid pixel coordinates and average MSE for one inference run

LD Datasets

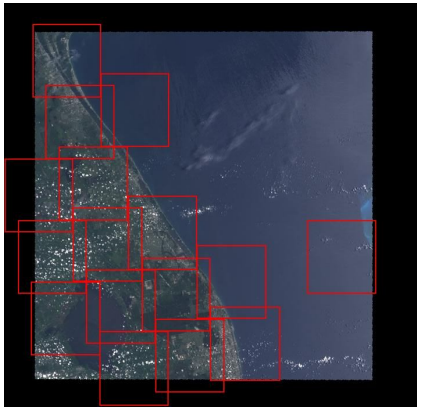
Landmark size: 25km x 25km



Train: Sentinel-2 mosaic



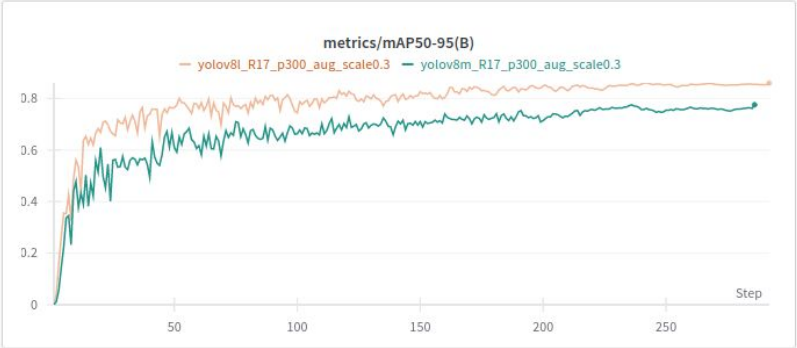
Validation: Rotated Landsat 8



Training Result - medium vs. large model

Augmentations:

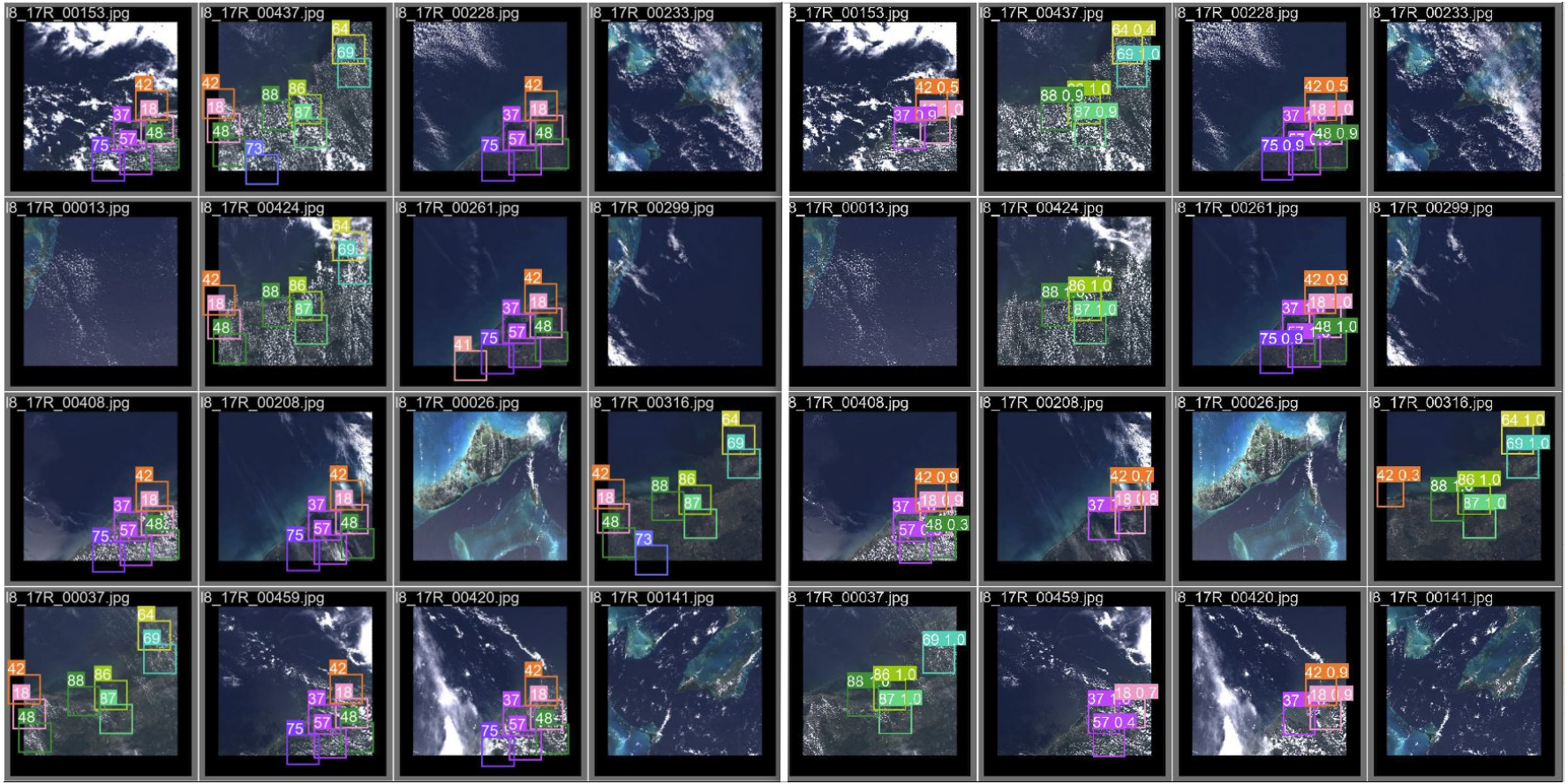
- Rotation
- Scale
- Perspective
- Translation
- Hsv color
- erasing

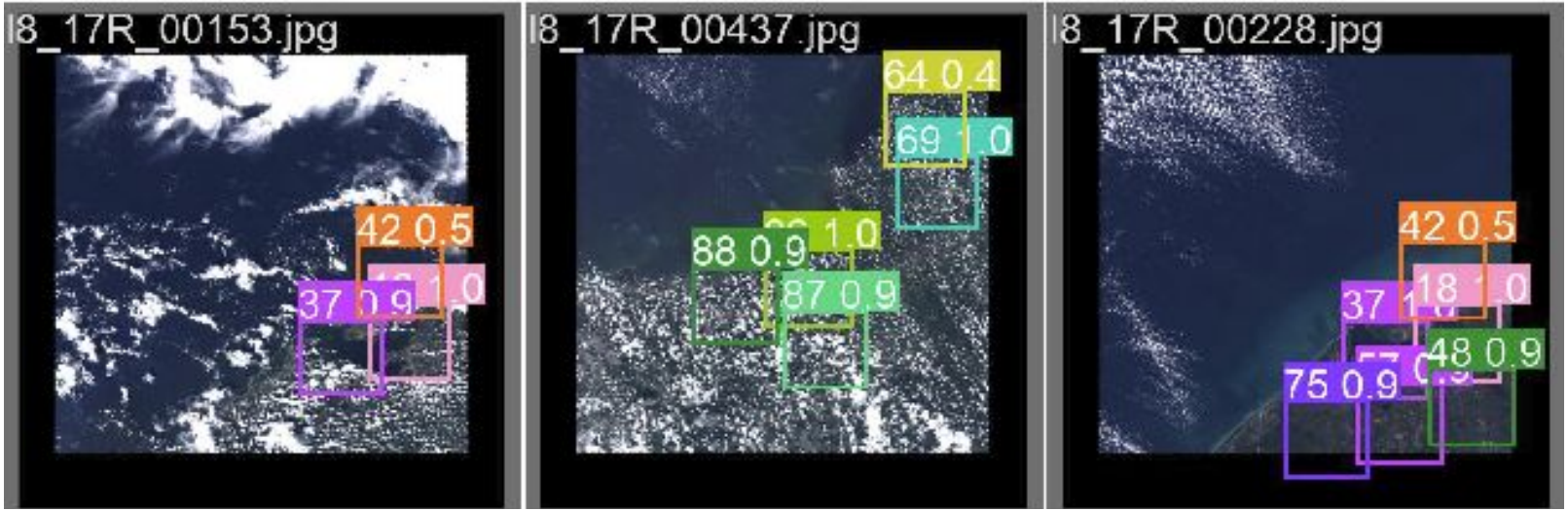


Training Result - Validation on Landsat

GT

Detected




8_17R_00153.jpg
42 0.5
37 0.9
8_17R_00437.jpg
64 0.4
69 1.0
88 0.9
87 0.9
8_17R_00228.jpg
42 0.5
37
18
48 0.9
75 0.9

Integrated testing

- Tested batch optimiser and coordinate transformation using landsat images displayed on the screen and image captured by testbed camera. - **Incorrect approach**
  - Mapped the centroids of the detected bounding boxes in the original Landsat ground truth to the image captured by the testbed camera.
  - Used these centroids to generate camera vectors pointing to landmark points
  - Ran the batch optimizer to determine attitude

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

2D Image Coordinates | Intrinsic properties (Optical Centre, scaling) | Extrinsic properties (Camera Rotation and translation) | 3D World Coordinates

Next steps

- Test the coordinate transformations using real world points seen from the reference camera to verify the camera unit vectors
- Test the batch optimizer with simulated world coordinates and camera matrices.

05/02/24

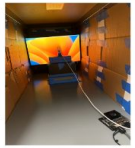
Demonstrating Visual-Inertial A&OD & On-Orbit Edge Computing

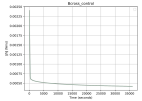
Progress summary

87 days before May 1st

Updates

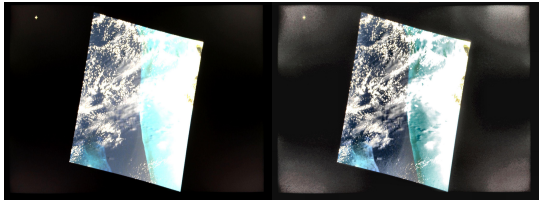
- Vision:
  - Data Pipeline automated
  - YOLO Training initial result
  - LD net inference integrated on Jetson
- Estimation:
  - Integrated Transformations and Static batch optimizer module into Vision Inference pipeline
- Testbed:
  - V1 testbed setup, completed calibration-undistortion code pipeline
  - Logging library for Jetson stats
- Simulator:
  - Preliminary MC analysis for B-cross control completed
- Completed integrated testing for V1





Blockers

- Camera finalization (avionics?)
- Differences between training pictures and test photos → need to test with the new camera



Weekly Plan

- Vision
  - Develop YOLO training plan & retrain LD models
  - Get high-performing LD model for one region w/o overfitting for V2 testbed
  - Complete integration of RC + LD inference on Jetson
- Estimation
  - Evaluate estimation accuracy of batch optimizer module
  - Validate the accuracy of the transformations module by comparing with ground truth.
  - Conduct rigorous integration testing to determine performance of the estimation pipeline

Interface dependencies

- Intra communication between Jetson and Pycubed
- Driver development for IMU, GPS, magnetorquer h-bridge

Vision

- **Dataset**
  - Data Pipeline (Image download + landmark annotation + convert to Yolo format) automated via shell script on workstation
- **Training**
  - **RC:**
    - Training efficientnet-b0 with multiple different hyperparameters, image sizes, dropouts etc.
  - **LD:**
    - Trained YOLO models of various sizes on data
    - Issue making detections on val set, likely because landmark boxes are too small/close together + val set too small, mAP close to zero after 100 epochs
    - Future training will be summarized here -> YOLO Training Report
- **Jetson**
  - LD inference class integrated
  - Testing script implemented for running inference through sample images + basic accuracy test
  - Sample inference data + pretrained 17R_nadir.pt model for V1

Automation Script

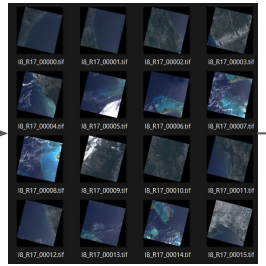
```
(myenv) (base) argus-vision@argus-vision:~/vision/VisionTrainingGround/DataPipeline$ ./run_pipeline.sh -h
Usage: ./run_pipeline.sh [options]

Options:
-b    BOUNDS        Geographic bounds (format: 'minLon minLat maxLon maxLat'). Default: '-84 24 -78 32'
-i    IDATE         Initial date (format: YYYY-MM-DD). Default: '2020-05-01'
-f    FDATE         Final date (format: YYYY-MM-DD). Default: '2023-12-31'
-l    LANDSAT       Landsat version. Default: 8
-m    MAXIMS        Maximum number of images. Default: 50
-s    SCALE         Scale. Default: 150
-w    BOX_WIDTH     Width of the boxes. Default: 100
-n    BOX_COUNT     Number of boxes. Default: 1000
-o    OUTPATH       Final output path. Default: '/home/argus-vision/vision/VisionTrainingGround/LD/datasets/17R_dataset'
-h                  Display this help and exit
```

```
(myenv) (base) argus-vision@argus-vision:~/vision/VisionTrainingGround/DataPipeline$ ./run_pipeline.sh
Generate download URLs: 100%| 50/50 [00:08<00:00,  6.17it/s]
Download Images: 100%| 50/50 [00:08<00:00,  6.07it/s]
Generate Saliency Maps: 100%| 50/50 [00:00<00:00, 63.57it/s]
Merging saliency maps..
Maps merged.
Create Landmark Bounding Boxes::   0%| 0/1 [00:00<?, ?it/s]
Window Size: 100
Number of Boxes: 1000
Create Landmark Bounding Boxes:: 100%| 1/1 [00:00<00:00,  1.85it/s]
Processing Images: 100%| 50/50 [00:03<00:00, 14.05it/s]
Generating Label Files: 100%| 50/50 [00:00<00:00, 1178.00it/s]
Dataset configuration saved to /home/argus-vision/vision/VisionTrainingGround/LD/datasets/17R_dataset/dataset.yaml
```

Data Pipeline Workflow

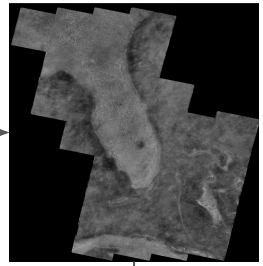
Google Earth Engine API
with control of
- Region
- Image count
- GSD / size
- Cloudcover
- etc..

Download images for specific region

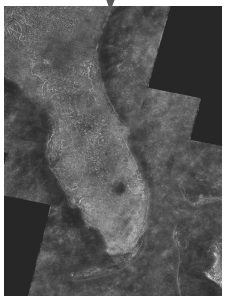


Create maps of local saliency for each image in region

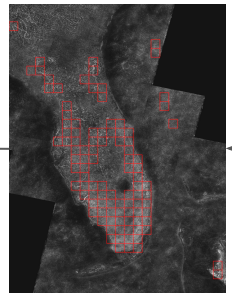
Merge and reproject maps of local saliency to get global saliency (of the region)



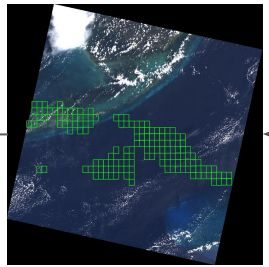
Crop to remove out of bounds area



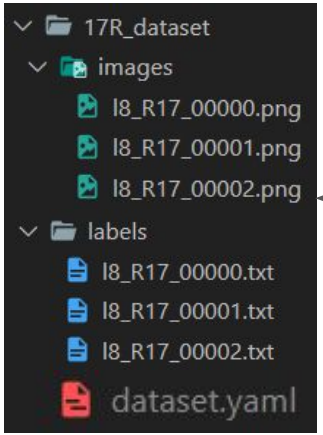
Scan through saliency map to find most salient boxes (landmarks)



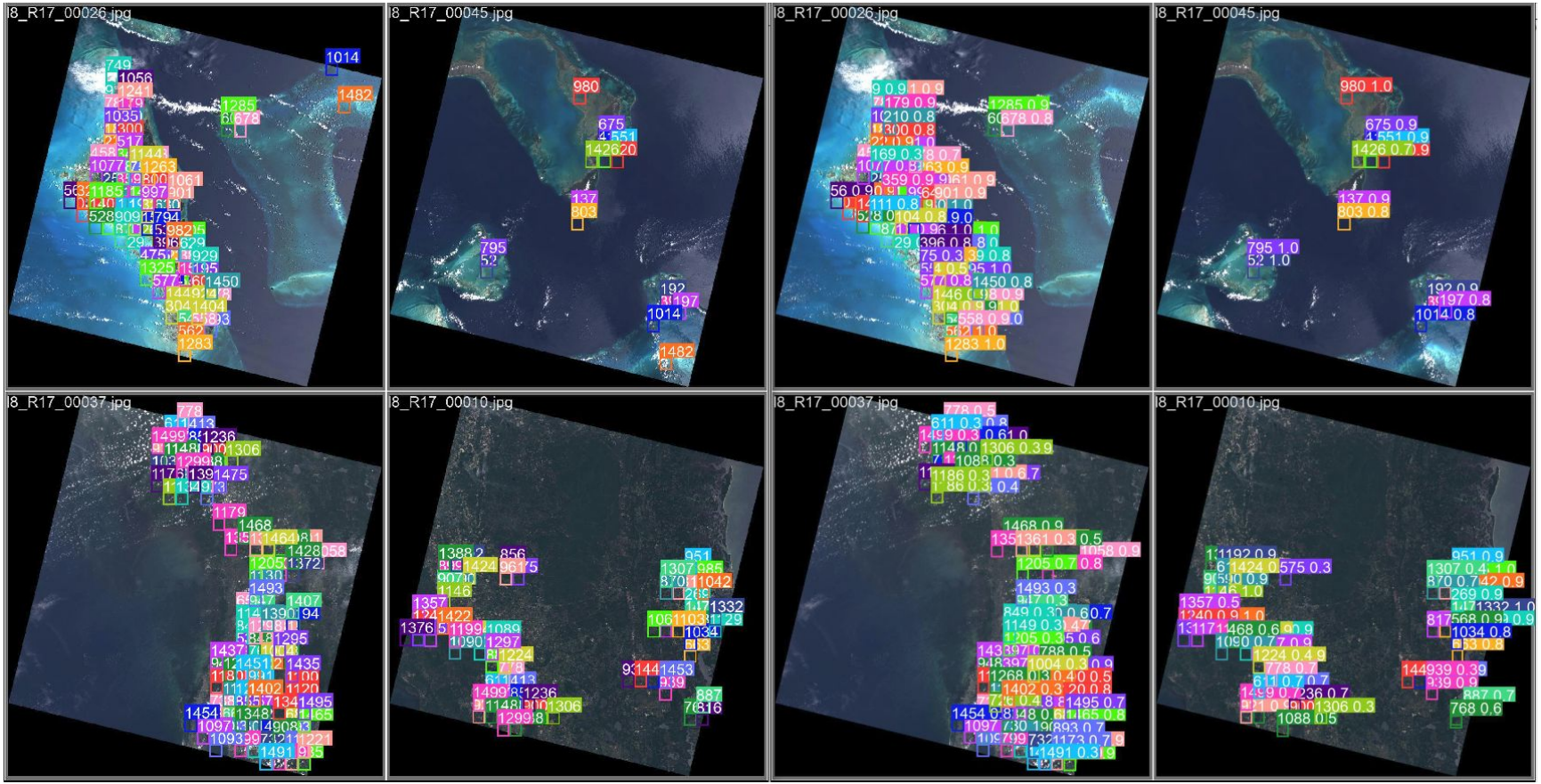
Assign landmarks to individual image



- Convert tif to png
- Assign class label for landmark
- Generate label file for each image
- Generate dataset config file



LD Training Result



Integrated testing - V1

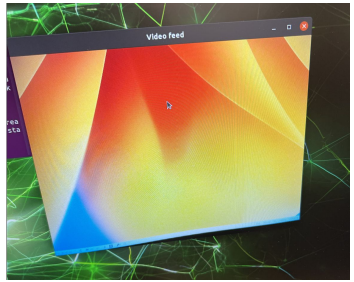
- Integrated vision/ML, coordinate transformations and batch optimization modules to execute estimation pipeline for V1 prototype
- Inference for landmark detection is done on existing images and result is converted to camera vector and ECI coordinates used for estimation.

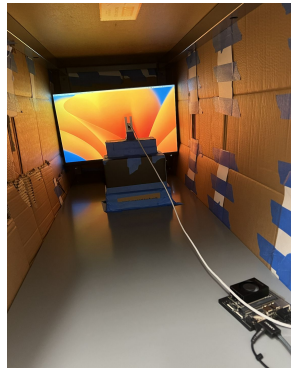
Next steps

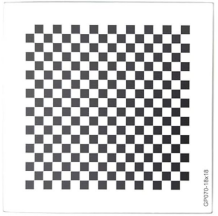
- Validate pixel coordinates to camera vector transformation ~ calibration
- Conduct entire pipeline test with images captured by camera
- Implement feature tracking between consecutive images, Batch LS

Dark room testbed

- All components are now relative to the ground truth (left edge of the table)
- Minimized error inside the setup by using spirit level, tape measure and caliper







Next steps

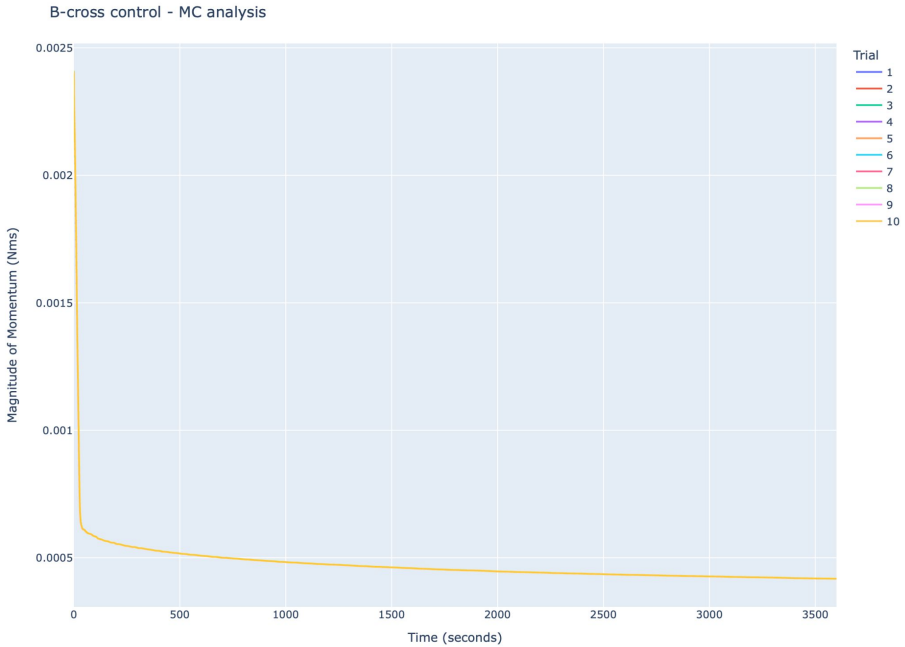
- Collaborate with vision testings and see if more modifications need to be made
- Merge the current setup as the starting point for V2 testing bed
- Look into how to match the projected image w/ the actual image

Simulator

- Features added:
  - Attitude and Orbit Sampling
  - Monte Carlo Analysis Setup
  - Logging and visualization support for MC analysis
- Testing:
  - Preliminary MC analysis of B-cross control
    - 10 runs, 60 minutes duration, 1 Hz control cycle, 10 Hz integrator cycle, Altitude [500 km - 600 km] SSO

Next steps:

- More testing with increased number of runs, different ranges for sun-synchronous orbits, different control and integrator sampling times
- Updates:
  - Sensor transformations based on location
  - Sun sensor model



B-cross_10runs_MC_Plot

Team schedule status → V2

V2	Vision Pipeline Validation and FSW kick-off	**Vision dev**	Data Engine Pipeline automation (--> Kyle)	Eddie Li
			Expand RC net to X regions	Jash Shah
			Selection filter (6 cameras)	Jash Shah
			Train LD net army	Eddie Li Haochen Zhang
			Sub-images and LD selector	Jash Shah
			Orbital pass simulation LD tests	Eddie Li Jash Shah Elakhya Nedumaran Atharv Pulapaka
		GNC dev	GNC Reqs & architecture	Ibrahima Sory Sow
			Tracking batch LS from LD	Ibrahima Sory Sow Atharv Pulapaka
			Sun sensor module development (calibration, sun vector)	Atharv Pulapaka
			(Sim/FSW) Update geometric model (sensor location, ...)	Elakhya Nedumaran Atharv Pulapaka
			A&OD integration testing w/ inference	Atharv Pulapaka Ibrahima Sory Sow
		FSW Jetson	FSW Jetson reqs	Ibrahima Sory Sow
			Health monitoring interface & external power logging	Sachit Goyal
			Test image compression algorithms on Jetson (Comms)	Sachit Goyal
			Develop and test interface Jetson <-> PyCubed	Sachit Goyal
			Camera interface FSW version	Tianxin Li
			Inference FSW version	Haochen Zhang Jash Shah
		FSW PyCubed	FSW PyCubed Reqs	Ibrahima Sory Sow
			Set-up I/O on PocketQube board w/ simulation (SIL)	Sachit Goyal
			MCM FSW SIL testing	Elakhya Nedumaran
		Darkroom Testbed	Image projection matching	Luyi Tang Tianxin Li
			V2 darkroom	Luyi Tang
			Calibrate all 6 new cameras	Tianxin Li
			Image downlink test, from Jetson --> PyC² --> Ground station	Sachit Goyal Ibrahima Sory Sow Atharv Pulapaka

Next steps

- Analyse results of V1 test
- Improve on dark room testbed
- Start development of V2

DD/MM/YY

Demonstrating Visual-Inertial A&OD & On-Orbit Edge Computing

Progress summary

100 days before May 1st

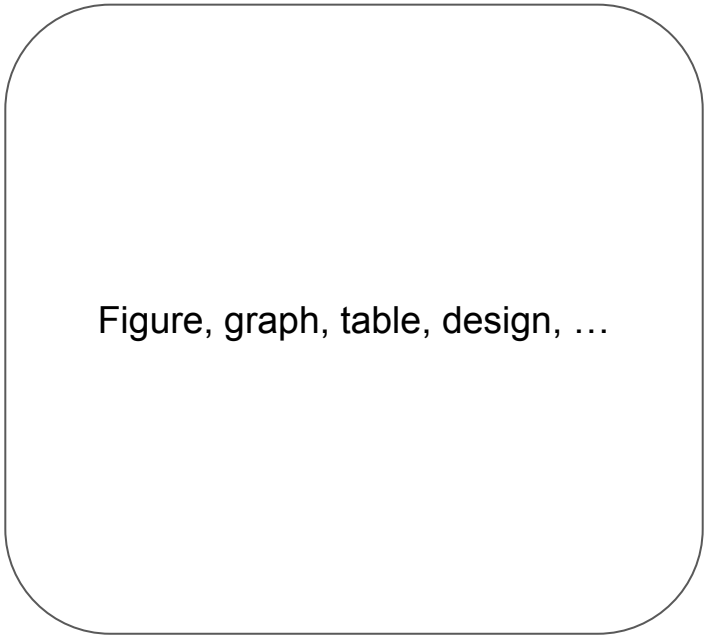
- Flight version: V1
- Metric improvement
  - Visual-Inertial A&OD accuracy: **X %** (vs X % last week)
  - LD inference error:
  - Landmark catalog size: X (vs … last week)
  - Camera error margin: X % (vs … last week)
- New features
  - Architecture change, new algorithm, test cases, hardware added, calibrated, tested, FSW deployed, …

Team schedule status

- Insert visual of schedule
- Next deliverables for next week (high-level)

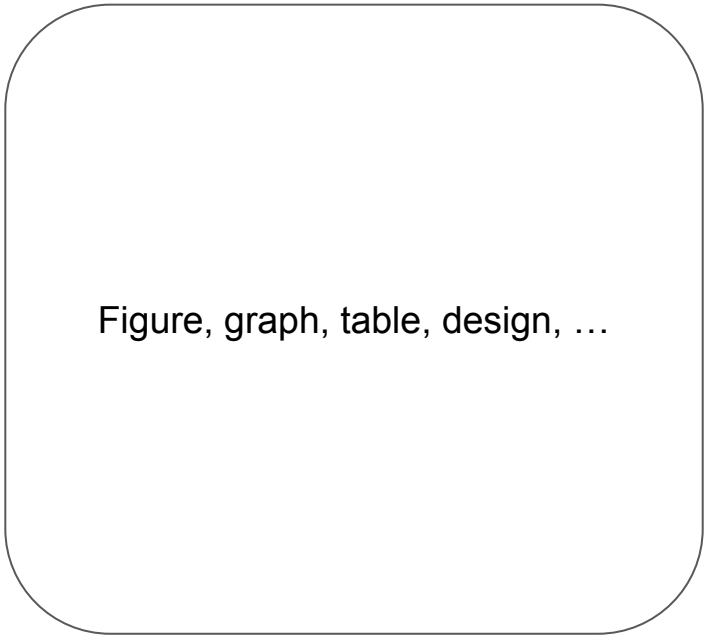
Topic 1 - Deep dive

- What? Problem you're solving
  - …
- Why is this important?
  - …
- How? Design, Algo, Methods, …
  - …
- Results? Improvements?
  - Ideally performance metric
  - …
- (Next on that)



Topic 2 - Deep dive

- What? Problem you're solving
  - …
- Why is this important?
  - …
- How? Design, Algo, Methods, …
  - …
- Results? Improvements?
  - Ideally performance metric
  - …
- (Next on that)



Message for other teams

- Put here any deadline, recommendations, bottleneck, interface change …

29/01/24

Demonstrating Visual-Inertial A&OD & On-Orbit Edge Computing

Progress summary

94 days before May 1st

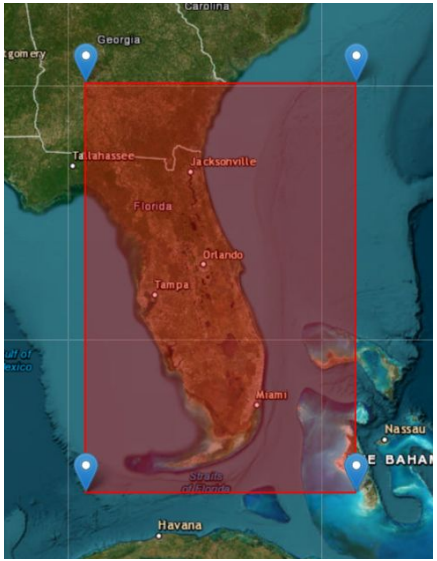
- Dark room testbed
  - Developed first iteration of dark-room test bed for V1 prototype
  - Completed single-camera calibration
  - Workstation and Jetson set-up
- Vision
  - Dataset pipeline changes & automated saliency-based landmark annotation
- Implemented all perturbations, sensor, and magnetorquer models in simulation
- Working prototype for A&OD (Batch optimization)
- Team changes: 1 out, 3 in
  - Task allocation and schedule updates

Vision

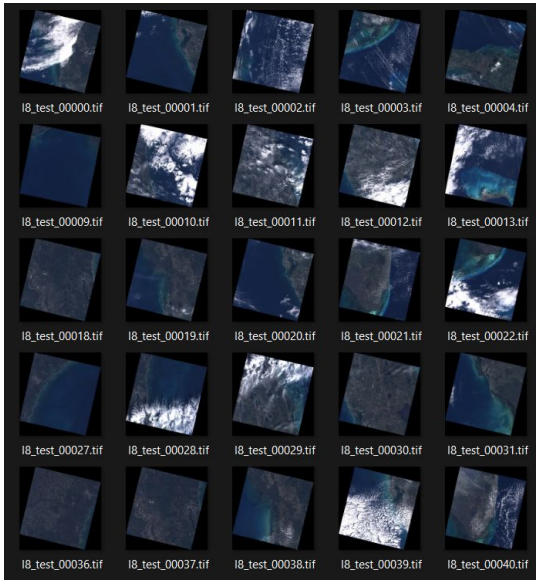
- New training plan for RC+LD structure
- Dataset Pipeline Design changes:
  - Data Access: USGS M2M API -> Google Earth Engine (more data)
  - Region Annotation: World Referencing System -> Military Grid System (for non-landsat images)
  - Preprocessing: Training image GSD 30m -> GSD 150mm (Camera lens 16mm -> 3.2mm for bigger FOV)
- Military Grid System based dataset download
- Automated saliency-based landmark annotation
- Small dataset for training & testing for V1
- **Next Steps**
  - Fully automated dataset pipeline on Workstation
  - Training RC net and LD net for V1
  - Refine landmark annotation (used less accurate code due to workstation fan issue)

V1 Dataset Prep

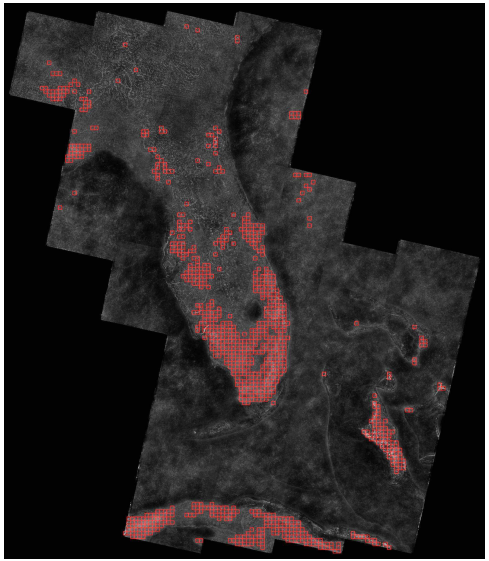
One Region: R17



~500 images GDS 150m



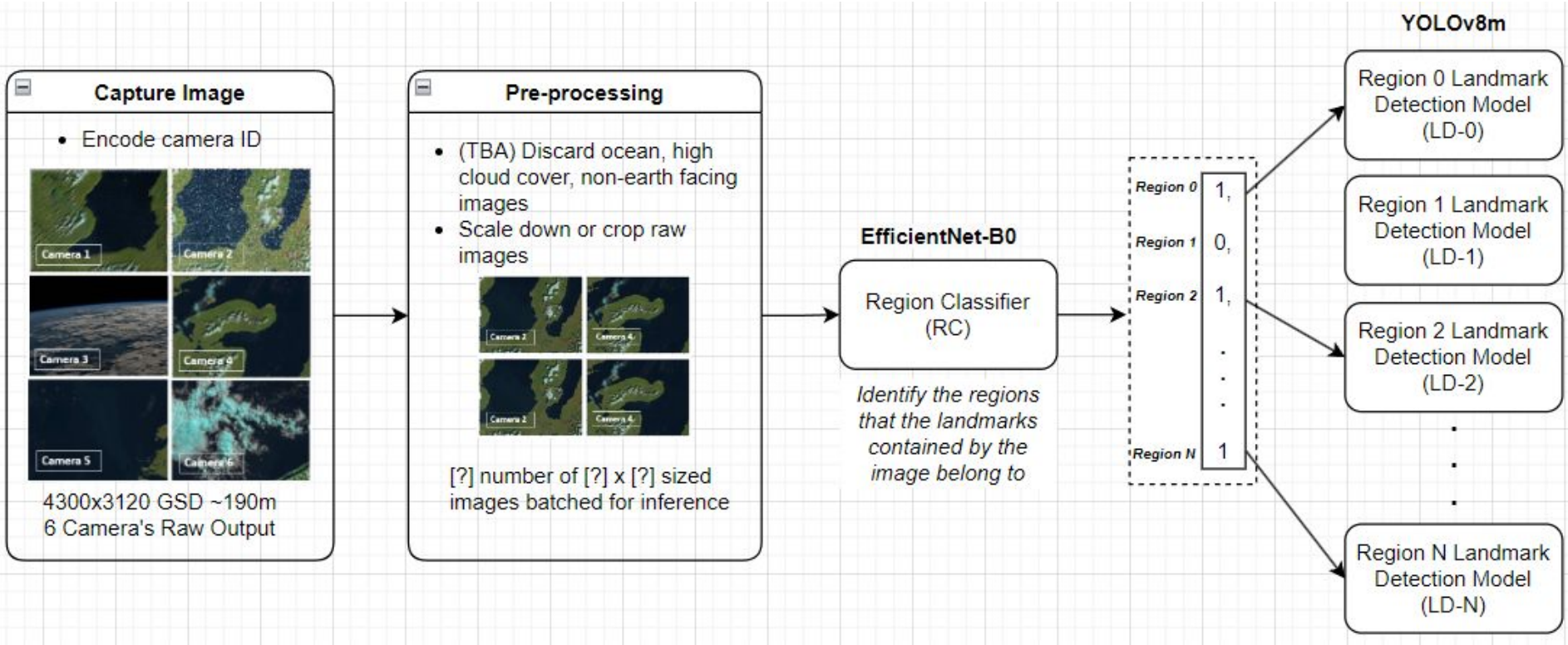
Top 1000 salient landmarks



Labeled with real lat/lon:

```
Top-Left Longitude,Top-Left Latitude,Bottom-Right Longitude,Bottom-Right Latitude
-79.489400090212,25.80369695391825,-79.44429409111412,25.751627180421764
-82.10554803788904,24.60609216349909,-82.06044203879117,24.554022390002604
-82.33107803337845,24.501952616506117,-82.28597203428056,24.449882843009632
-82.24086603518268,24.554022390002604,-82.1957600360848,24.501952616506117
-79.17365809652684,25.543348086435824,-79.12855209742895,25.49127831293934
-79.12855209742895,25.49127831293934,-79.08344609833108,25.439208539442852
-82.28597203428056,24.501952616506117,-82.24086603518268,24.449882843009632
-82.28597203428056,24.554022390002604,-82.24086603518268,24.501952616506117
-82.15065403698692,24.60609216349909,-82.10554803788904,24.554022390002604
-79.39918809201623,26.27232491538662,-79.35408209291836,26.22025514189013
-79.57961208840776,24.345743296016664,-79.53450608930987,24.293673522520177
-82.1957600360848,24.554022390002604,-82.15065403698692,24.501952616506117
```

New Inference/Training Plan

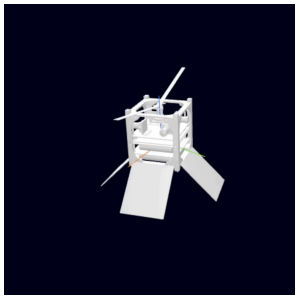


Simulation

Developing simulator to test and validate detumbling, A&OD, and FSW algorithms and support Software-in-the-loop (SIL) testing.

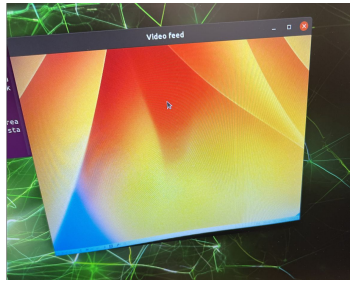
Progress

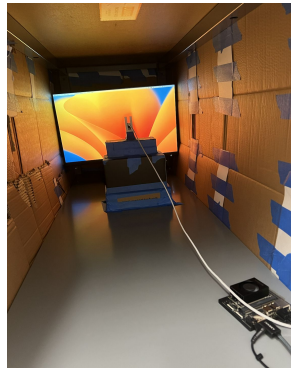
- Incorporated atmospheric drag and third body effects in orbital perturbations.
- Added sensor model for gyro, magnetometer, sun sensor and GPS.
- Implemented magnetorquer model.
- Implemented B-cross control along with unit tests.

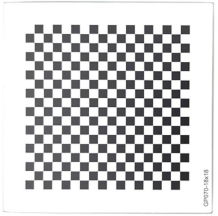


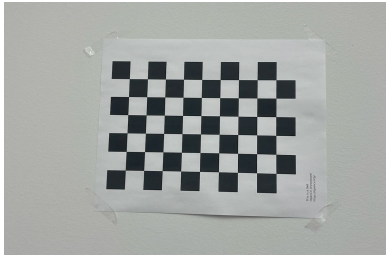
Dark room testbed

- Initial setup for V1 pipeline testing in dark-room environment along with associated hardware
- Progress
  - Camera position and parameters adjustments for field-of-view on projector
  - Camera interface software
  - Single-camera calibration pipeline and generated camera intrinsic and extrinsic matrices and distortion parameters
- Next steps
  - Calibrate all 6 cameras, w/ chessboard
  - Research towards multi-camera calibration (mock-up Cubesat)
  - Research and experiments (?): match images captured from screen to actual image in dataset
  - Report testbed errors
  - Start V2 dark room design









Testbed hardware / FSW - next steps

- Initial discussion w/ Comms for telemetry and command
- Jetson <-> PyCubed intercommunication driver requirements w/ Avionics
- Next steps
  - Dark room test bed
    - Metrics data logging (latency, power, cpu, gpu, ..) on Jetson for V1
  - PyCubed board (acquired one for GNC usage)
    - Set-up basic I/O w/ simulation for Software-in-the-loop testing
    - Learn Pycubed stack and implement basic flight software
  - Inference V1
    - Test inference pipeline on V1 testbed (Jetson)

Team schedule status

A	B	C	D	E	F	G
Flight Version	**Version name**	**Module**	**Tasks**	**Assignment**	22 Jan	29 Jan
		Single-camera calibration	Camera interface (Jetson)	Atharv Pulapaka		
			Calibration software pipeline 1 camera	Tianxin Li		
			Calibrate 6 cameras (id them) and store parameters	Tianxin Li		
			Multi-calibration research	Tianxin Li		
			Image pre-processing & img storage pipeline	Atharv Pulapaka		
			Body vector transfo + landmark retrieval	Atharv Pulapaka		
		Dataset	Set-up workstation	Sachit Goyal		
			Donwload dataset on workstation + annotate	Eddie Li Jash Shah		
			Saliency-based annotation automation	Eddie Li		
			TIF -> lat-long to ECI landmark database	Atharv Pulapaka		
		Training system	Training plan	Eddie Li		
			Train RC net (all regions)	Jash Shah		
			Train LD net (1 region)	Haochen Zhang Eddie Li		
		Inference	Implement inference pipeline (Jetson)	Haochen Zhang		
			Integrate w/ camera interface	Haochen Zhang Atharv Pulapaka		
			Test RC & LD pipeline on Jetson	Haochen Zhang Eddie Li		
			Conversion to body vector + landmark retrieval from the database	Atharv Pulapaka		
V1	**Nadir-only Payload Pipeline Validation**	**Batch optimization**	Prototype optimization	Ibrahima Sory Sow		
			Simulation validation + test on Jetson	Ibrahima Sory Sow		
			Integration test w/ inference	Ibrahima Sory Sow		
		Simulation	Perturbations and sensor models	Elakhya Nedumaran Atharv Pulapaka		
			Vision measurement model	Atharv Pulapaka Ibrahima Sory Sow		
		MCM (Magnetic Control)	Initial Bcross Monte-Carlo	Elakhya Nedumaran		
			FSW version ready for SIL	Elakhya Nedumaran		
		Test-bed	Install dark room set-up (covers, tent, whatever)	Luyi Tang		
			Get screen	Luyi Tang		
			Camera mount	Luyi Tang		
			Research & experiments - image matching	Luyi Tang		
			Report testbed errors	Luyi Tang		
			V2 darkroom design	Luyi Tang Ibrahima Sory Sow Luyi Tang		
			Metrics data logging (latency, power, cpu, gpu, ..)	Sachit Goyal		
		PyC	Familiarize w/ current PocketQube board	Sachit Goyal		
			Set-up basic I/O w/ simulation (SIL prep)	Sachit Goyal		
			Intercommunication design dev (PyC <-> Jetson)	Sachit Goyal		

Next steps

- Finalize V1 prototype integration and testing before this Friday
  - Test of the entire pipeline in the dark room from pixel to A&O estimates
  - Report preliminary results
    - Pixel errors for landmark detection
    - A&O accuracy
    - Testbed source of errors
- V2 refinement based on learnings

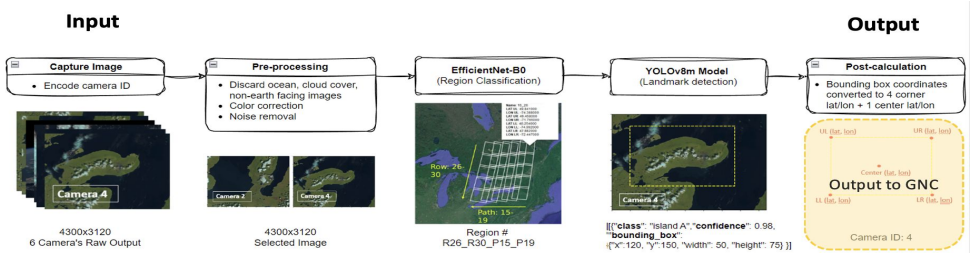
22/01/24

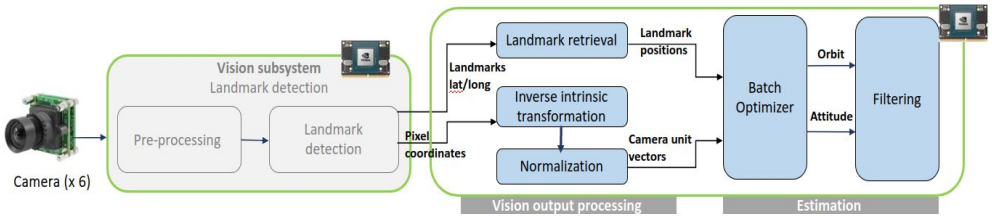
Demonstrating Visual-Inertial A&OD & On-Orbit Edge Computing

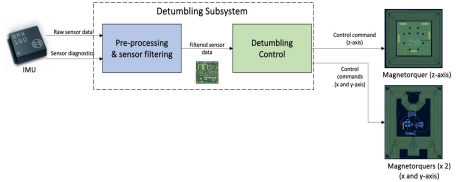
Payload-GNC-FSW - Team merging

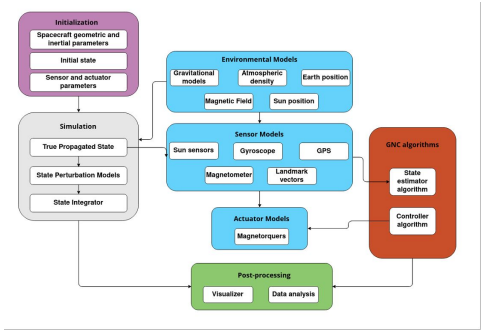
- PDR
- Initial work on AD/OD Pipeline
- GNC hardware selection
- Datasets
- Orbit analysis
- Simulator v1
- FSW Functional architecture
- Detumbling control
- Helmholtz Cage Design
  - Cancelled ⇒ Dark room testbed

⇒ Lots of cross-development with Avionics & Comms









100 days before May 1st

Plan for the semester

- Payload-GNC-FSW coupled development
- **Two–week development cycles** with predetermined milestones, plan and deliverables
  - Design iterative prototypes of full pipeline to reach final project completion
  - Each prototype is **fully-functional** and includes development, unit testing and integration testing with hardware
- Why?
  - Quick feedback and learning on the whole development process (both HW and SW)
  - Hands-on approach w/ hardware and boards
  - Consistent validation and metric improvement
  - Motivation

100 days before May 1st

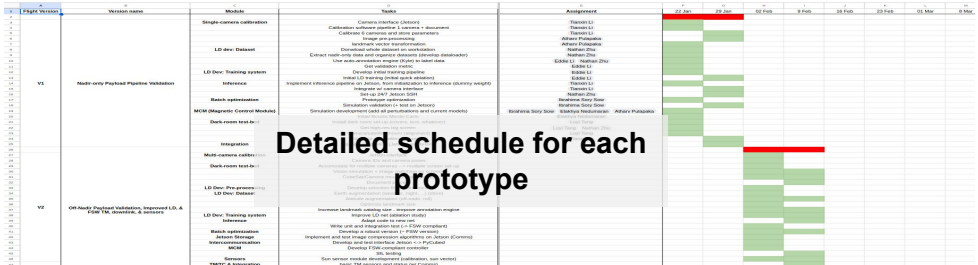
Schedule overview for the semester

Successive development of versioned prototypes (Vn). V1, V2, V3 are fully detailed. V4+ will be expanded as we incorporate feedback. Details on prototype and schedule are here:

- https://docs.google.com/document/d/1MIU3D8B_4Isp-4LFPRzEAeG_gtccRu3iFakApqQPfzs/edit?usp=sharing
- https://docs.google.com/spreadsheets/d/1F0aAKpx0vbFHra7poJnUTnJOS3EivXlNLYzwoGcZlVg/edit?usp=sharing

Flight Version		22 Jan	29 Jan	02 Feb	9 Feb	16 Feb	23 Feb	01 Mar	8 Mar	15 Mar	22 Mar	29 Mar	5 Apr	12 Apr	19 Apr	26 Apr	3 May
V1	Nadir-only Payload Pipeline Validation																
V2	Off-Nadir Payload Validation, Improved LD, & FSW TM, downlink, & sensors																
V3	A&OD updates, early V&V Infrastructure, Power control																
V4	A&OD updates, Autonomy and CD&H dev & testing																
V5	OD dev updates freeze, Autonomy and CD&H, Distributed CD&H, (over-the-air) OTA updates																
V6	FSW A&O freeze, V&V, CD&H, Satellite characterization, Failure Management																
V7	Final satellite assembly, FSW integration, and testing, documentation + buffer																
V8	Final satellite assembly, FSW integration, and testing, documentation + buffer																

Schedule for all prototypes

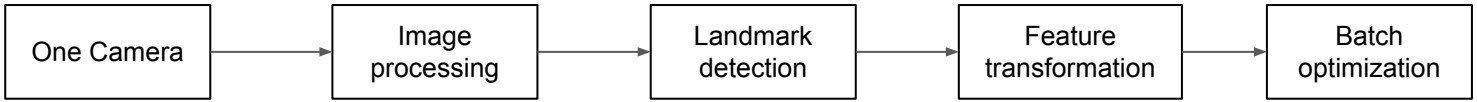


Detailed schedule for each prototype

100 days before May 1st

V1: Nadir-only Payload Validation

- Validation of A&OD pipeline in **dark-room testbed** with **a single-calibrated camera** on **Jetson**.
- Camera captures an image from a well-positioned screen, which is pre-processed and LD net is applied. Necessary transformations then batch optimization to determine attitude and orbit.
- **Simulation of a LandSat pass and report average accuracies**



- Determine landmark catalog size
- Obtain average number of landmarks per images (number of features)
- Define validation metric for landmark detection net
- Determine attitude and orbit estimation accuracy
- Build dark-room testbed in the lab, implement on testbed and find error margins
- Run MCM simulation and analyse performance

100 days before May 1st

V1: Nadir-only Payload Validation ⇒ Feb 2

Module	Tasks	Assignment
Single-camera calibration	Camera interface (Jetson)	Tianxin Li
	Calibration software pipeline 1 camera + document	Tianxin Li
	Calibrate 6 cameras and store parameters	Tianxin Li
	Image pre-processing	Atharv Pulapaka
	landmark vector transformation	Atharv Pulapaka
LD dev: Dataset	Donwload whole dataset on workstation	Nathan Zhu
	Extract nadir-only data and organize datasets (develop dataloader)	Nathan Zhu
	Use auto-annotation engine (Kyle) to label data	Eddie Li Nathan Zhu
	Get validation metric	Eddie Li
LD Dev: Training system	Develop initial training pipeline	Eddie Li
	Initial LD training (initial quick ablation)	Eddie Li
Inference	Implement inference pipeline on Jetson, from initialization to inference (dummy weight)	Tianxin Li
	Integrate w/ camera interface	Tianxin Li
	Set-up 24/7 Jetson SSH	Nathan Zhu
Batch optimization	Prototype optimization	Ibrahima Sory Sow
	Simulation validation (+ test on Jetson)	Ibrahima Sory Sow
MCM (Magnetic Control Module)	Simulation development (add all perturbations and current models)	Ibrahima Sory Sow Elakhya Nedumaran Atharv Pulapaka
	Initial Bcross Monte-Carlo	Elakhya Nedumaran
Dark-room test-bed	Install dark room set-up (covers, tent, whatever)	Luyi Tang
	Get high-res big screen	Luyi Tang Nathan Zhu
	Camera/satellite mount (alignment)	Luyi Tang
	Integrate with calibration set-up	Tianxin Li
Integration	Pipeline integration (Jetson) and testing	