



Developing a Computer Vision System for Autonomous Satellite Maneuvering

Andrew Harris, PhD
Senior Systems Engineer
SCOUT Space Inc.



What We're Talking about Here

1. SCOUT's pose estimation approach and competition performance
2. How we did and lessons learned
3. Pose estimation demo
4. Future work + challenges for the field

SCOUT: Perception for Spacecraft

- SCOUT is developing perception systems to enable the next-generation of autonomous satellites to **avoid debris** and **keep space safe**
- Two major domains:
 - **Close range, proximity operations**
 - Long-range, space domain awareness



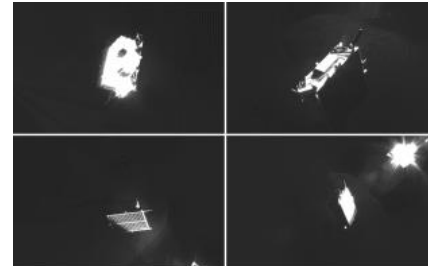
Space Domain Vision: Is It Hard, or Just Different?

Challenges:

- Extremely data-limited
- Sensitive to safety, correctness issues
- Relatively compute constrained

Prospects:

- Low-clutter, typically simple backgrounds
- “Knowable” lighting conditions, dynamics
- Well-defined shapes (usually)

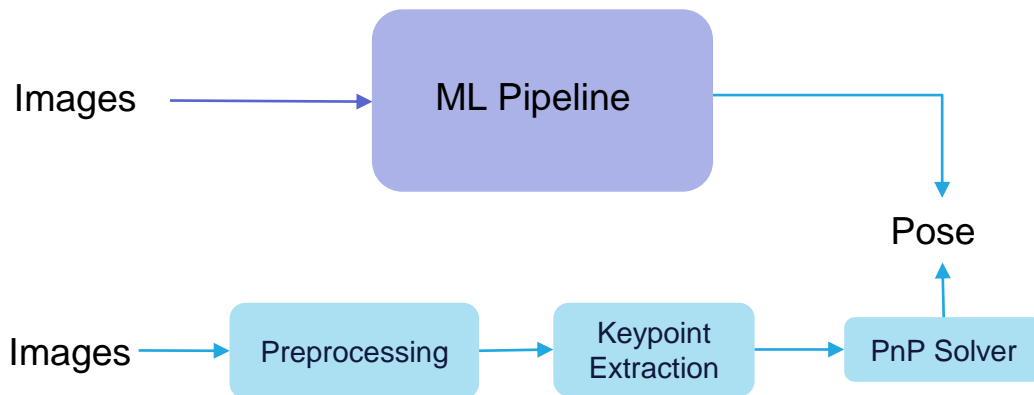


sunlamp

SCOUT vs. Traditional Approaches

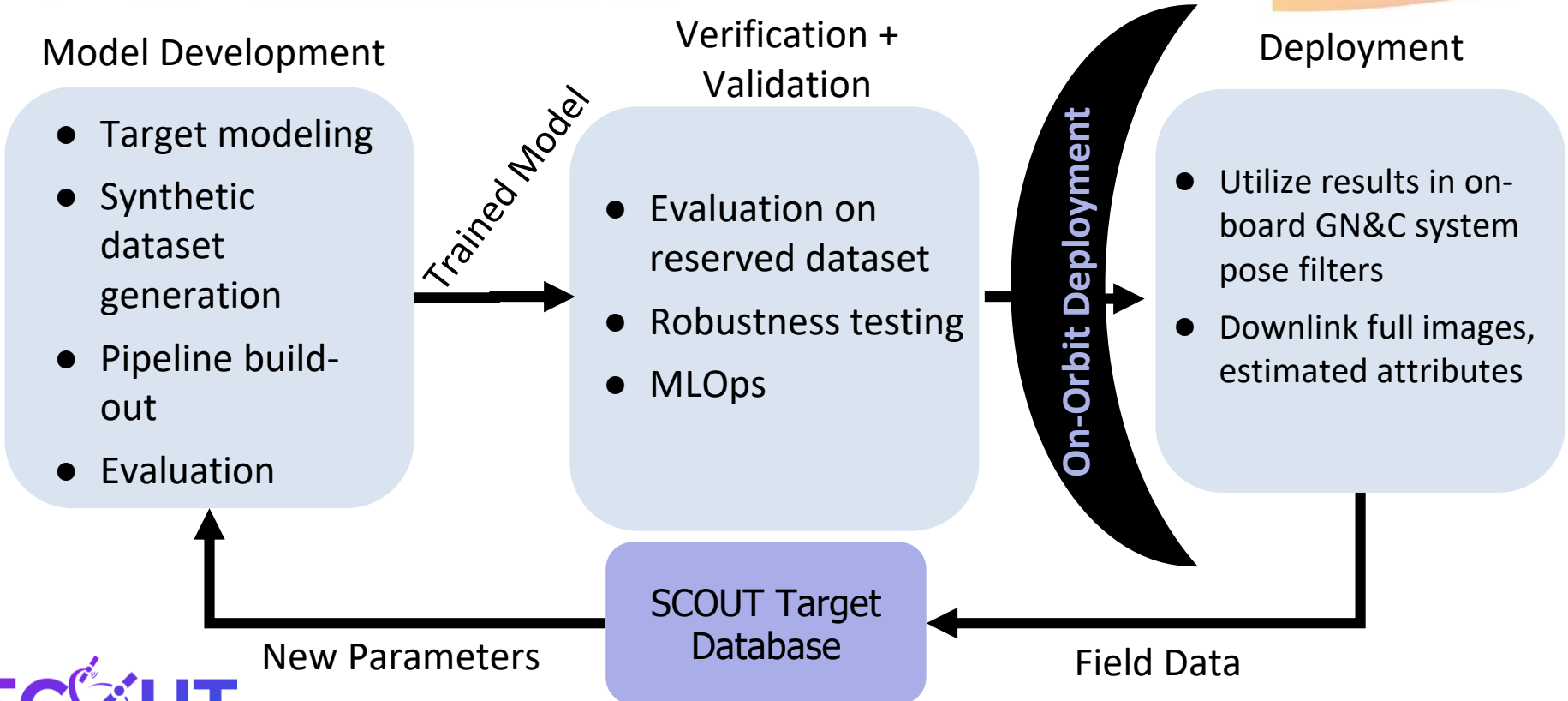
- SCOUT's proximity operations navigation solution leverages ML-driven pose estimation
- Many tradeoffs vs. 'traditional' approach
 - Fewer parameters to tune
 - Generalizability across lighting domains
- Aim is to improve runtime + sensitivity vs traditional approaches

SCOUT's approach



Traditional Pose Estimation

SCOUT: Perception Systems

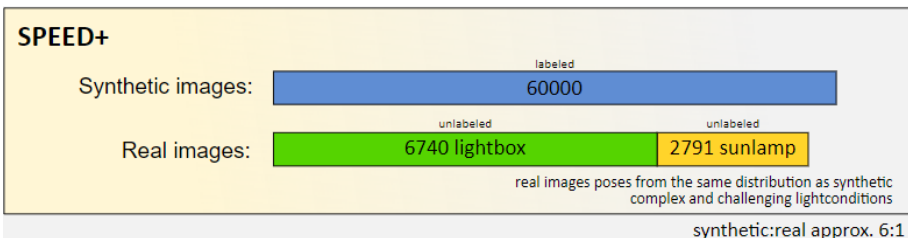


SCOUT: The Long Road to Pose



ESA Kelvin Pose Estimation Challenge: A Motivating Problem

- ESA competition to improve image-driven pose estimation technology
 - Inspired by the Prisma formation flight mission (right)
 - SPEED+ dataset: ~10k physical images from SLAB testbed, 60k simulated images from SLAB simulator
- Scored based on sum of position and attitude estimation error



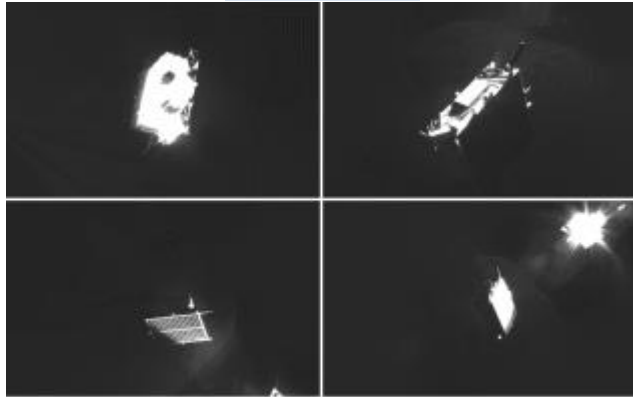
$$\text{score}_{\text{orientation}}^{(i)} = \begin{cases} 0, & \text{if } \text{err}_{\text{orientation}}^{(i)} < 0.169^\circ \\ \text{err}_{\text{orientation}}^{(i)}, & \text{else} \end{cases}$$

$$\text{score}_{\text{position}}^{(i)} = \begin{cases} 0, & \text{if } \text{err}_{\text{position}}^{(i)} < 0.002173 \\ \text{err}_{\text{position}}^{(i)}, & \text{else} \end{cases}$$

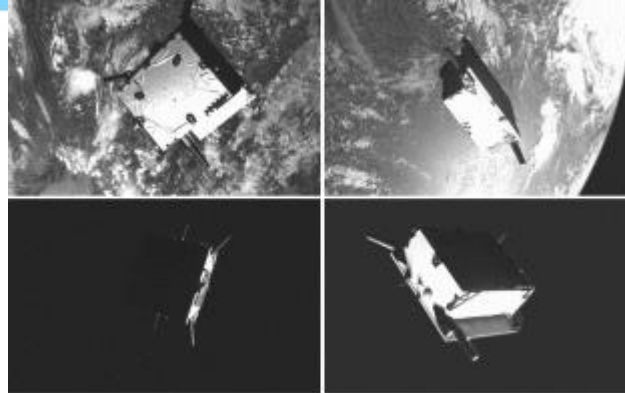
$$\text{score}_{\text{pose}}^{(i)} = \text{score}_{\text{orientation}}^{(i)} + \text{score}_{\text{position}}^{(i)}$$

ESA Kelvin Pose Estimation Challenge: Closing the Domain Gap

sunlamp

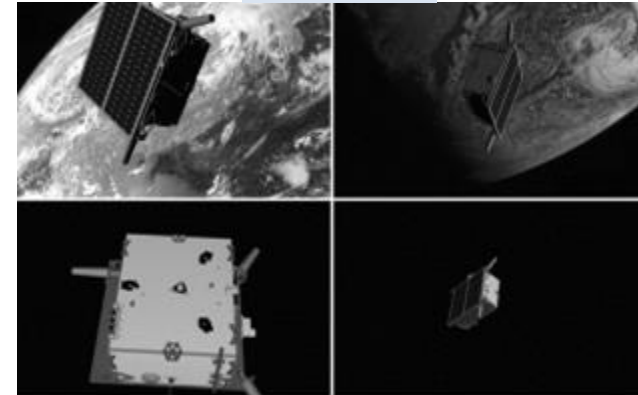


sunlamp

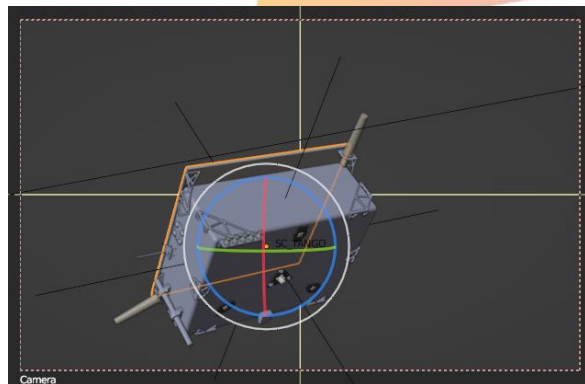


lightbox

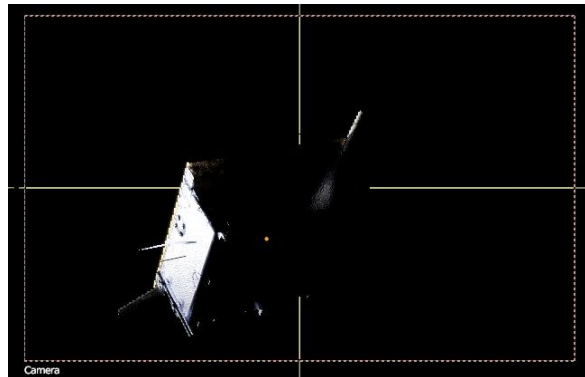
synthetic



First Attempt: Blender



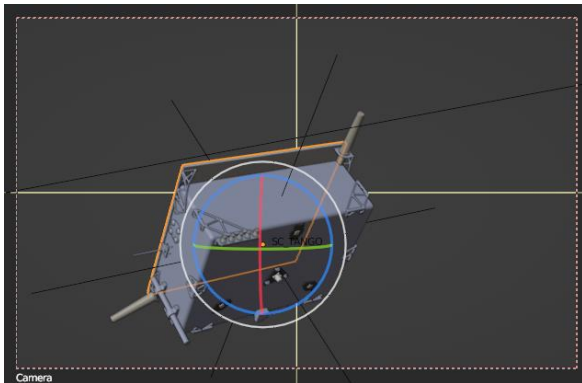
CAD model of a spacecraft (Tango)



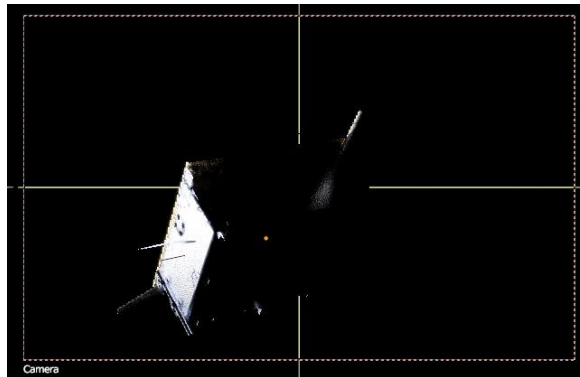
Photorealistic render of Tango

- Blender used as scene generation suite of choice
- Naïve / unrefined Earth, S/C parameters
- No noise, star background; only resolution challenges

"Are the Synthetics Realistic?"

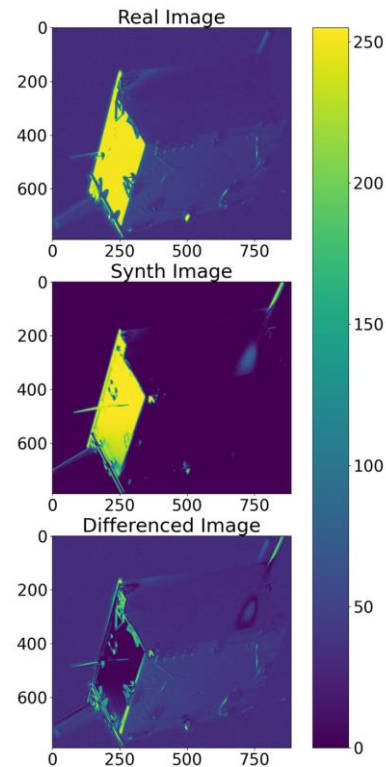


Blender model of Tango from CAD



Photorealistic render of Tango

- Render pipeline generally *looks* good, but is not necessarily realistic
- Higher reflectivity than lab Tango
- Some component mismatches from real mock-up (see right)
- Missing diffuse back-reflection



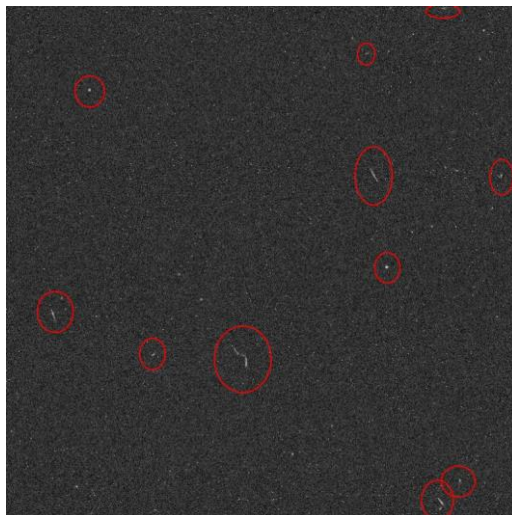
Comparison vs. lightbox image 11

Does it matter, and how do we fix it?

Improving Realism: Things to Consider

- Lighting
 - Lighting conditions change rapidly on-orbit
 - Streaking/exposure
 - Sun angle / glint
- Blur sources (focus, motion)
- Detector noise
 - Shot, dark current
 - Cosmic rays

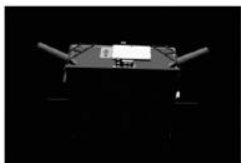
Right: Photograph of an Iridium flare against star background



Left: Long exposure showing multiple suspected cosmic ray hits

Stanford Space Rendezvous Laboratory: Augmenting Data with Synthetic Noise

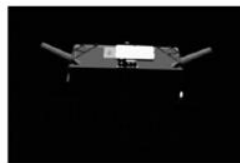
Augmentation	Commands
Brightness & Contrast	RandomBrightnessContrast
Random Erase (Zhong et al., 2020)	CoarseDropout
Sun Flare	RandomSunFlare
Blur	OneOf(MotionBlur, MedianBlur, GlassBlur)
Noise	OneOf(GaussNoise, ISONoise)



original



RandomBrightnessContrast



MedianBlur



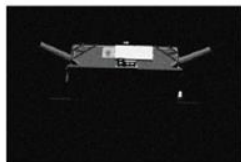
MotionBlur



GlassBlur



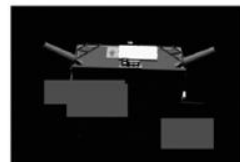
style augmentation



GaussNoise



ISONoise



CoarseDropout



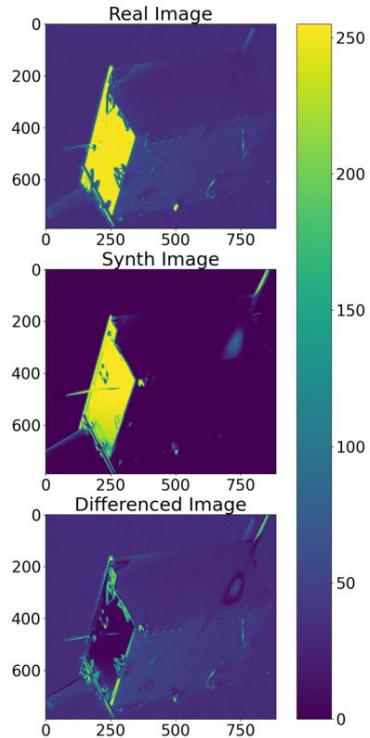
RandomSunFlare

Stanford Space Rendezvous Laboratory: Improving Performance with Synthetic Data

Config.	Source	lightbox				sunlamp			
		IoU [-]	E_T [m]	E_R [°]	E_{pose}^* [-]	IoU [-]	E_T [m]	E_R [°]	E_{pose}^* [-]
Baseline	E	0.853	0.518	24.678	0.509	0.867	0.641	47.893	0.937
	H	-	0.506	21.994	0.465	-	0.735	47.546	0.955
+ Random Erase	E	0.811	0.756	24.168	0.534	0.510	2.766	79.232	1.771
	H	-	0.665	22.544	0.494	-	2.295	80.778	1.744
+ Sun Flare	E	0.892	0.314	11.670	0.252	0.825	0.875	33.239	0.709
	H	-	0.347	10.018	0.230	-	0.722	31.504	0.661
+ Style Aug.	E	0.918	0.175	8.004	0.169	0.919	0.225	12.433	0.254
	H	-	0.271	6.479	0.158	-	0.307	11.065	0.245

Competition Results





Lightbox

Rank	Team Name	norm. err pose	norm. err rot	Best Score
1	TangoUnchained	0.0179	0.0556	0.073498689
16	SCOUT Inc	0.0909	0.8357	0.926615725
35	baseline	0.3686	2.2038	2.572462691

Sunlamp

Rank	Team Name	norm. err pose	norm. err rot	Best Score
1	lava1302	0.0113	0.0476	0.058860147
14	SCOUT Inc.	0.0832	1.0750	1.158212043
35	baseline	0.3736	2.2002	2.573856284

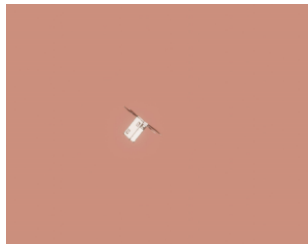
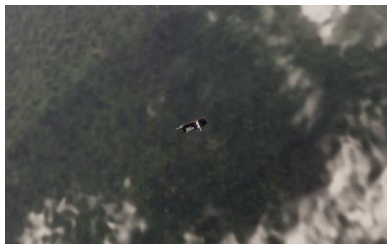
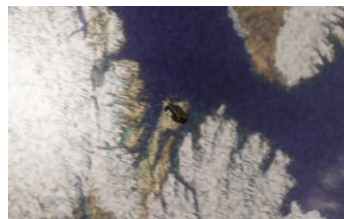
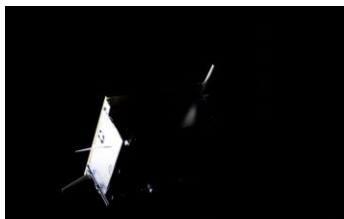
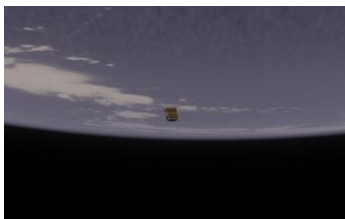
¹ Burkhardt Z., Spessert, E., West, S., Gallucci, S., et al. "Trajectory Planning for a Proximity Operations Flyby Operation on the Tenzing Mission." In AAS Guidance and Control Conference 2022. AAS-22-155. February 2022.

Demo of SV-50 Inference



Creating a Model to Generate Synthetic Data: SCOUT's System's Capabilities

- Renders at ~3000 images/hour
 - Specific trajectory
 - Randomized ranges/pose/backgrounds
 - Earth or stars background for realistic image generation
 - Color/randomized image background for general training



SCOUT Render Pipeline Demo

The screenshot displays a Jupyter Notebook environment with several tabs open: `camera_process.py 2`, `camera_manager_process.py`, `cameras.log`, `inferencer.log`, `camera_process.py`, `inference.py`, `competition_inference.py`, and `simulated_camera.py x`. The active notebook is titled `scout_vision` and contains a `SimulatedCamera` class with an `init` method.

The `Scout Command Interface` window shows a log of interactions:

```
2022-12-08 02:46:07 Connected to Server!
2022-12-08 02:46:05 Connected to TLM Server!
2022-12-08 02:57:51 Sending Command:
2022-12-08 02:57:50 Sending TLM
Camera_Take_Snapshot ttag=1670486270.21
id: 1670486271
args:3 1 1
2022-12-08 02:57:51 Received TLM:
RelativePositionVector_m:
-17000.21
-16255.56
184983.98
2022-12-08 02:57:51 Received Response:
Camera_Take_Snapshot 0.72
id: 1670486271 0.36
status: OK -0.46
First image id: 190 0.37
```

Below the log, there are buttons for `Send`, `Clear`, `Exit`, `Connect`, and `Connect TLM`. A text input field for `Camera_Take_Snapshot` contains the value `311`. The interface also displays a satellite image of Earth with a purple overlay indicating a field of view or sensor range.

The bottom panel shows the `TERMINAL` output:

```
2022-12-07 23:57:52,369.INFO.updating response tracker!
2022-12-07 23:57:52,369.INFO.attempting to update the snap index file
2022-12-07 23:57:52,370.INFO.file id returned from snap command: 190 from b'\xbe\x00\x00\x00\x00\x00\x00'
==> scout_vision/logs/inferencer.log <==
2022-12-07 23:57:52,370.INFO.Attempting prediction on /mnt/ssd/scout/data/output/00000000190
==> scout_vision/logs/cameras.log <==
2022-12-07 23:57:52,371.INFO.checking 1 for data!
2022-12-07 23:57:52,372.INFO.received response with message id: 1670486271
2022-12-07 23:57:52,372.INFO.updating response tracker!
2022-12-07 23:57:52,374.INFO.attempting to update the snap index file
```

Demo of SV-50: Real-Time Inference

```
Translation(xyz, mm): [-35.37, 52.33, 257446.02]  
Rotation(quat): [0.03, 0.04, 0.02, 1.00]
```



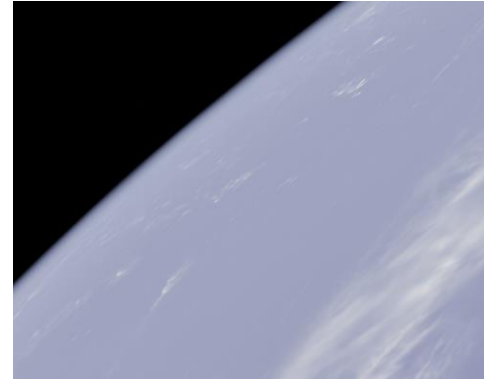
Integrating Real Data and Conclusions



Where's the Real Data?

- Images are big vs. space downlink pipes
 - 9.6 kilobaud connections are very common
- Emphasizing an iterative approach for data collection campaigns
 - Tenzig (2021): Noise + lens parameters
 - Near-term missions (2024): Target and SDA images in different lighting conditions

*Right:
Actual
photo from
SV-50 on
Tenzig*



*Left:
Simulated
image from
SCOUT's
synthetics*

Conclusions

- Space is hard, not impossible
- Synthetics are an inevitable part of space-based ML systems, so we have to learn to live with them
- ML pipelines seem to generalize well from synthetics to physical data in the lab, given synthetic images with similar noise + aberrations
- Standards and references for verification and testing are **essential** for deploying future machine vision systems in space (and on Earth!)

- Flight experiments! 3 (!!!) SCOUT systems will fly in 2024
- Automated verification + validation pipelines
- Learning pose estimation for arbitrary or damaged spacecraft
(=unknown geometry a-priori)

Synthetic Data: Tutorials and Examples

Synthetic Data Resources

SCOUT: Spacesight

<https://spacesight.scout.space/>

Space ML

<https://spaceml.org/>

Synthetic Data Tutorial

<https://bit.ly/synth-data>

SLAB Resources

SLAB Website

<https://slab.stanford.edu/>

SLAB Pose Estimation Paper

[arXiv:2203.04275v1](https://arxiv.org/abs/2203.04275v1)

Robotic Testbed for Models

[arXiv:2108.05529v2](https://arxiv.org/abs/2108.05529v2)

The logo for SCOUT features the word "SCOUT" in a bold, white, sans-serif font. The letter "O" is replaced by a stylized white satellite in orbit, with two solar panels extending from its sides. The background is a vibrant night sky showing the Milky Way galaxy in shades of purple, blue, and orange, with silhouettes of trees at the bottom.

SCOUT

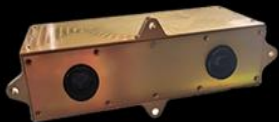
INFO@SCOUT.SPACE

Backup Material



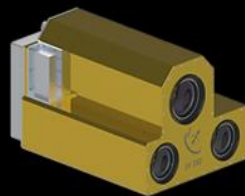
SCOUT: Perception Systems

SCOUT-VISION



Relative Navigation, Satellite Servicing

SV-250



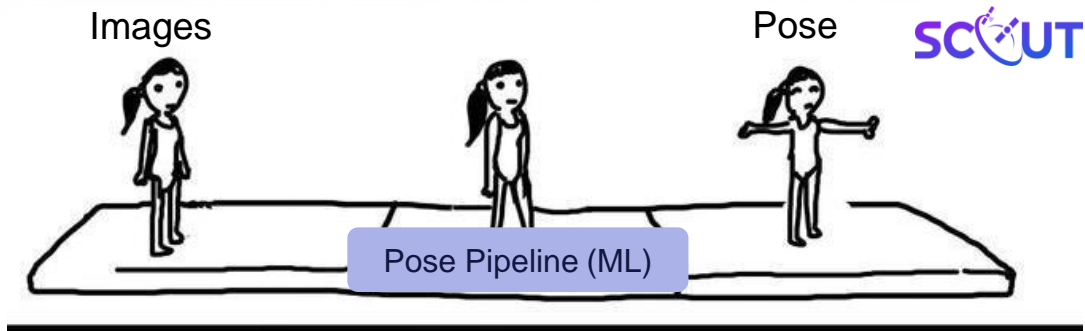
Local Situational Awareness

NITE-OWL

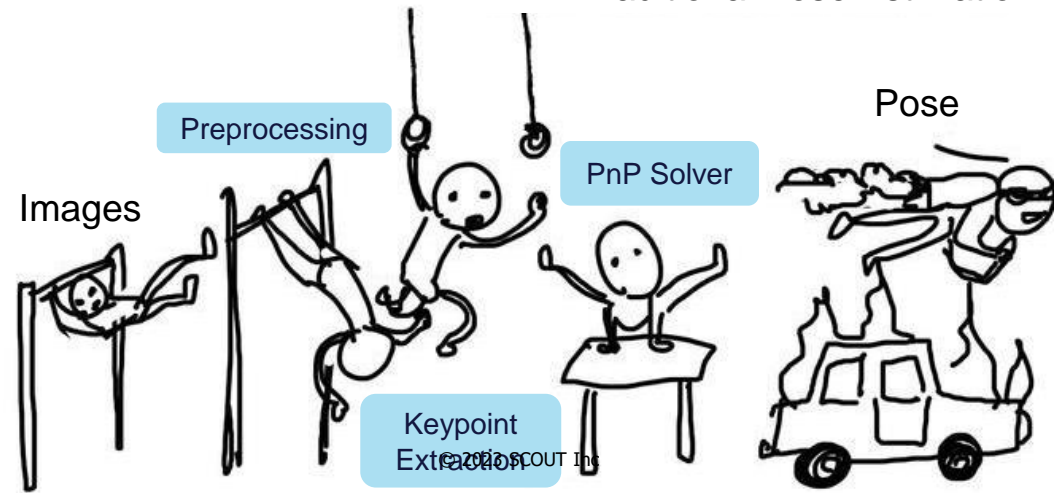


Long-Range & Cislunar

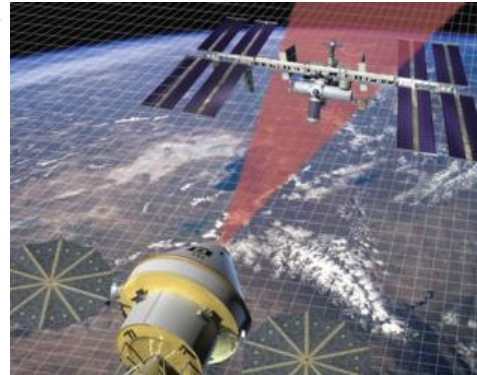
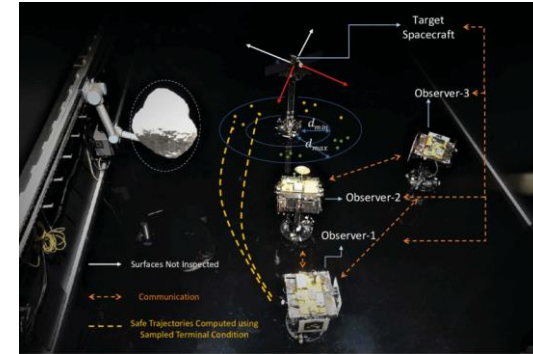
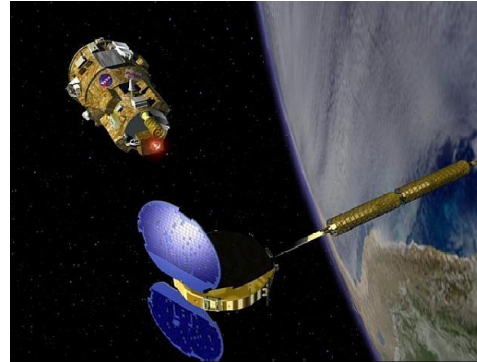
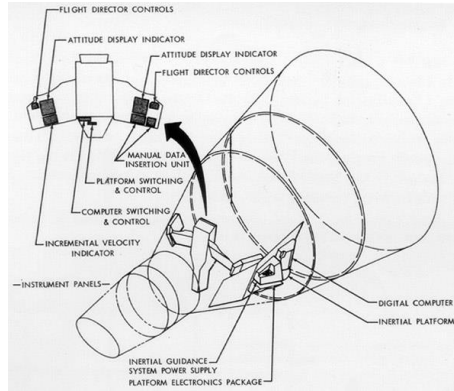
SCOUT vs. Traditional Approaches



Traditional Pose Estimation



Previous Work: Proximity Guidance



SCOUT: Remote-Sensing in Space On-Orbit Spacecraft Inspection

The SCOUT interface is displayed against a starry space background. On the left, a 'Lens Settings' panel includes sliders for Focal Length (250 mm), Focal Distance (87.5 m), Circle of Confusion (0.03 mm), and N (f/N aperture) (10). A 'USE HYPERFOCUS' toggle is checked. Below are sections for 'Environment Settings' and 'Sensor Settings', and a green 'Render' button. On the right, a 3D visualization shows a camera with a 5.591-degree FOV. A target satellite is at 100m. The field of view is divided into Near Focus Distance (61.672m) and Far Focus Distance (150.551m). A green box indicates the Hyperfocal Distance is 208.58m. A data panel on the far right lists various optical parameters.

Parameter	Value
Near Focus Distance (m)	61.672
Far Focus Distance (m)	150.551
Hyperfocal Distance (m)	208.583
Depth of Field (m)	88.879
FOV (degrees)	5.591
Pixel Pitch XY (um)	4.883
Sensor Diag. XY (mm)	12.207
Image Size (um)	5000.00
Image Size (px)	1024.00

Autonomous Edge System Considerations: Quality of Data

1. Data continuity: the system must be able to handle drop-outs in detection from CV model
2. Data reliability: the system needs physically-informed models to mitigate false-positive or extremely inaccurate CV measurements
3. SCOUT has developed estimation filters which propagate target position/pose based on existing data and equations of motion across signal dropouts and which improve effective relative navigation accuracy

Evaluating Trustworthiness of Autonomous Machine Learning Systems

- Your system operates as expected in the simulated environment, how to improve confidence levels that system will operate as expected when deployed to real-world environment

ESA Kelvin Pose Estimation Challenge: Loss Function/Scoring

$$\text{err}_{\text{position}}^{(i)} = \frac{|r_{gt}^{(i)} - r_{est}^{(i)}|_2}{|r_{gt}^{(i)}|_2}$$

$$\text{err}_{\text{orientation}}^{(i)} = 2 \cdot \arccos(|\langle q_{est}^{(i)}, q_{gt}^{(i)} \rangle|)$$

$$\text{score}_{\text{position}}^{(i)} = \begin{cases} 0, & \text{if } \text{err}_{\text{position}}^{(i)} < 0.002173 \\ \text{err}_{\text{position}}^{(i)}, & \text{else} \end{cases}$$

$$\text{score}_{\text{orientation}}^{(i)} = \begin{cases} 0, & \text{if } \text{err}_{\text{orientation}}^{(i)} < 0.169^\circ \\ \text{err}_{\text{orientation}}^{(i)}, & \text{else} \end{cases}$$

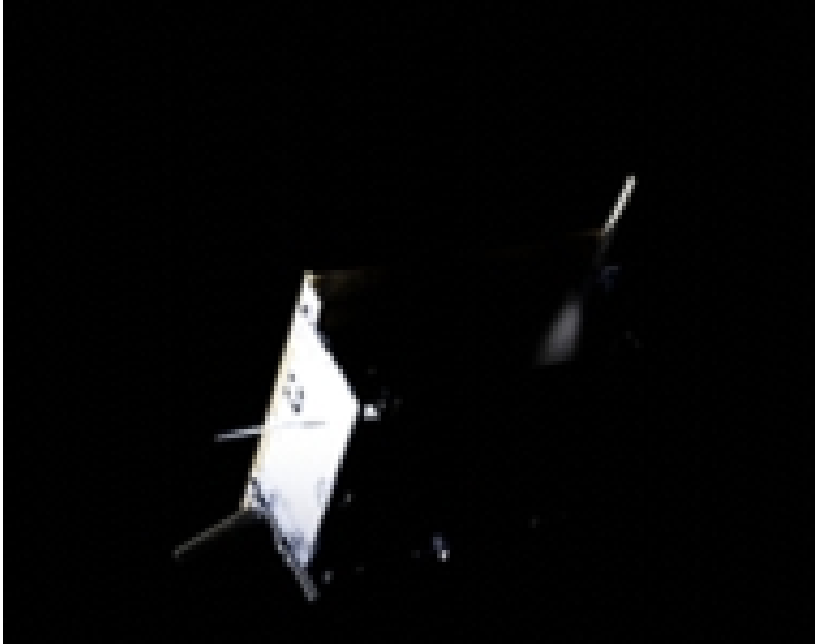
$$\text{score}_{\text{pose}}^{(i)} = \text{score}_{\text{orientation}}^{(i)} + \text{score}_{\text{position}}^{(i)}$$

$$\text{score} = \frac{1}{N} \sum_{i=1}^N \text{score}_{\text{pose}}^{(i)}$$

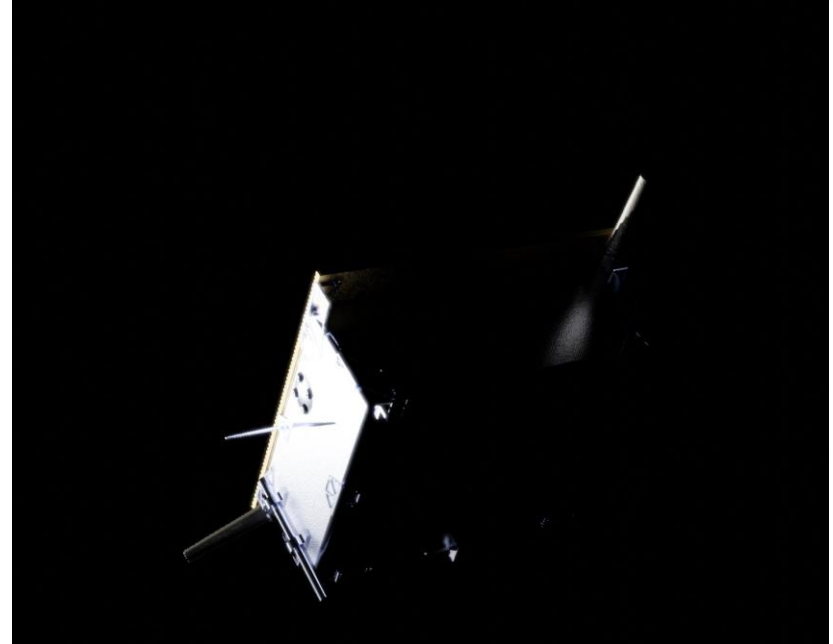
Loss Function Challenges

- Lots of spacecraft, including Tango, exhibit various symmetries
 - "off by 90/180" errors are extremely easy to come by
- Range is a major factor
 - <300 m: Fully resolved, maximum danger
 - >300 m: maybe partially resolved (can't get orientation), less dangerous
 - >2 km: Non-resolved, dynamics less linear

Determining Dataset Requirements: Resolution



Lower resolution image
of Tango spacecraft



Higher resolution image
of Tango spacecraft

Determining Dataset Requirements: Resolution vs. Exposure

Pixel Size

4.25 μm



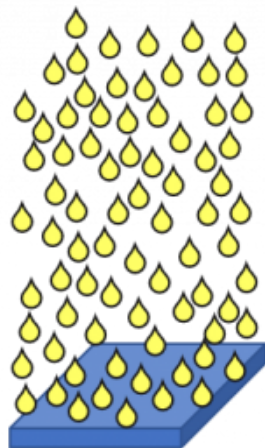
12,000e⁻

6.5 μm



45,000e⁻
(~3.8X more e⁻)

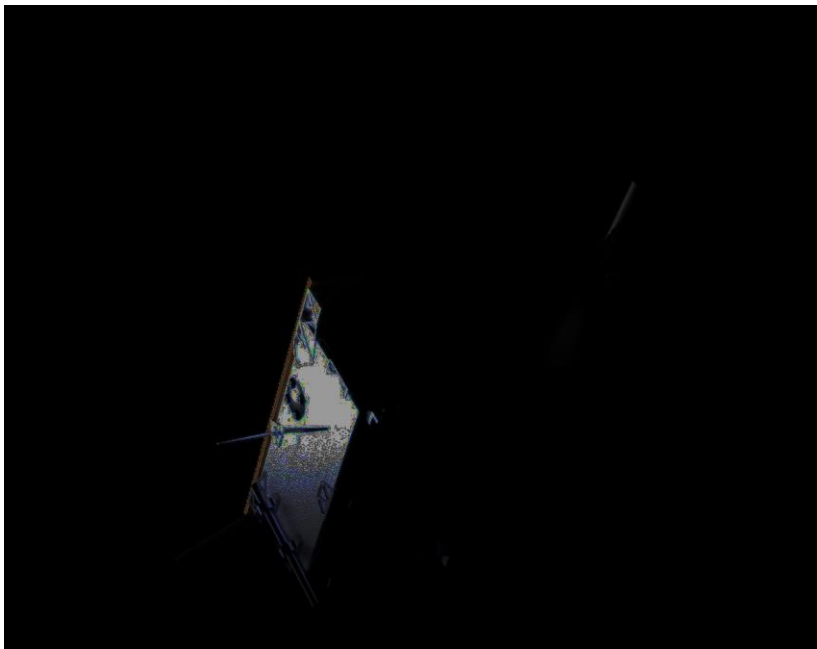
11 μm



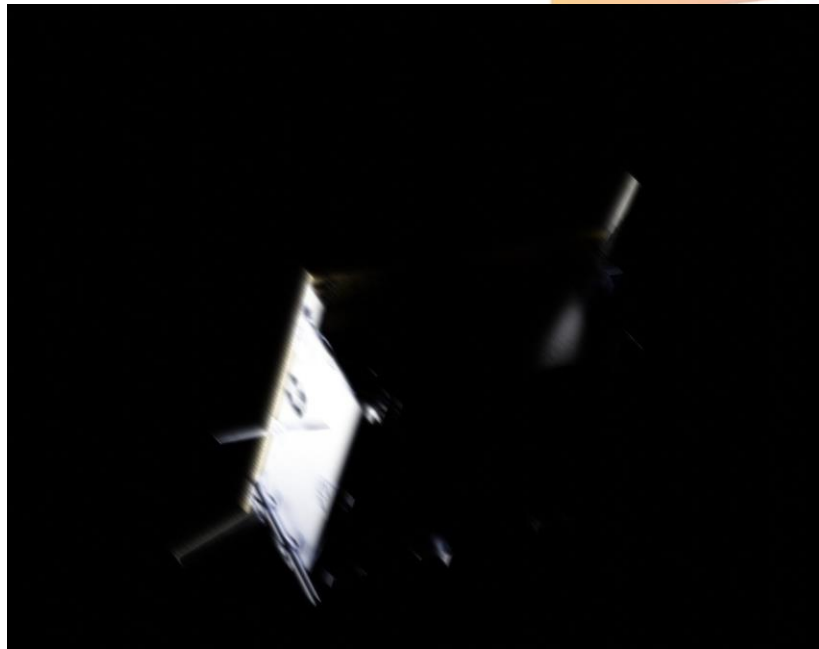
80,000e⁻
(~6.3X more e⁻)

Full-Well
Capacity

Determining Dataset Requirements: Fidelity and Resolution – Exposure Time



Underexposed image of
Tango spacecraft



Properly exposed image of Tango
spacecraft exhibiting motion blurring