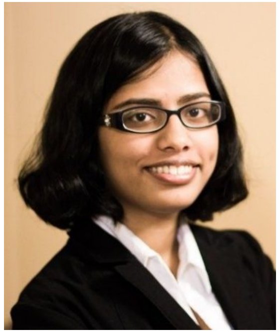
NASA Space Robotics Challenge Phase 2

TeamL3

TeamL3

7 engineers and researchers

Working part-time together remotely from California, Canada, Japan and Australia



Apeksha Budhkar

Louis-Jerome Burtz

Fabian Dubois

Nathaniel Guy

Ashish Kumar

Ilya Kuzovkin

Evan Smal

With the collaboration and support of
OffWorld Inc. and Mission Control Space Services Inc.

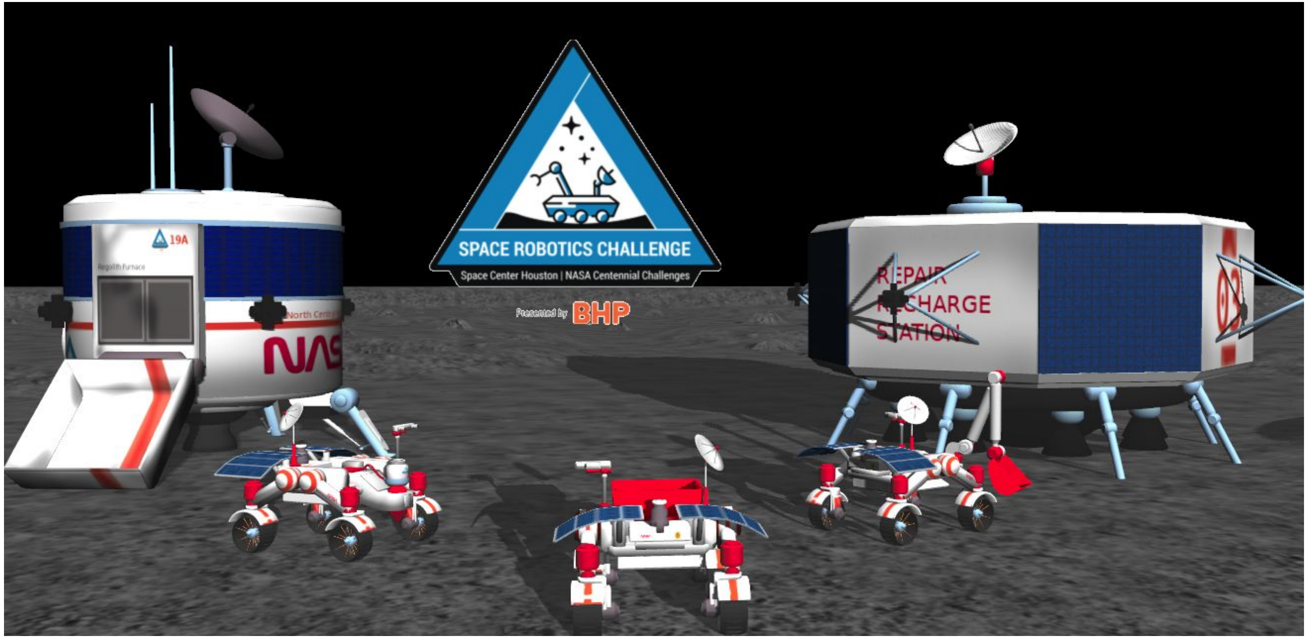
The Competition - NASA Space Robotics Challenge Phase 2

NASA Centennial Challenge
[Video Presentation]

1 Million USD total prize purse
>100 teams, >450 participants signed up

Fully autonomous:
Rovers explore the Lunar South Pole for resources

Fully virtual :
Simulated environment + simulated rovers and landers (Gazebo/ROS)



2020: Qualification phase: 22 finalists announced on Jan 15th 2021
2021: Final phase: winners announced on Sept 28th 2021

NASA CENTENNIAL CHALLENGES

Environment

Computer Simulated Lunar Surface (Gazebo/ROS)

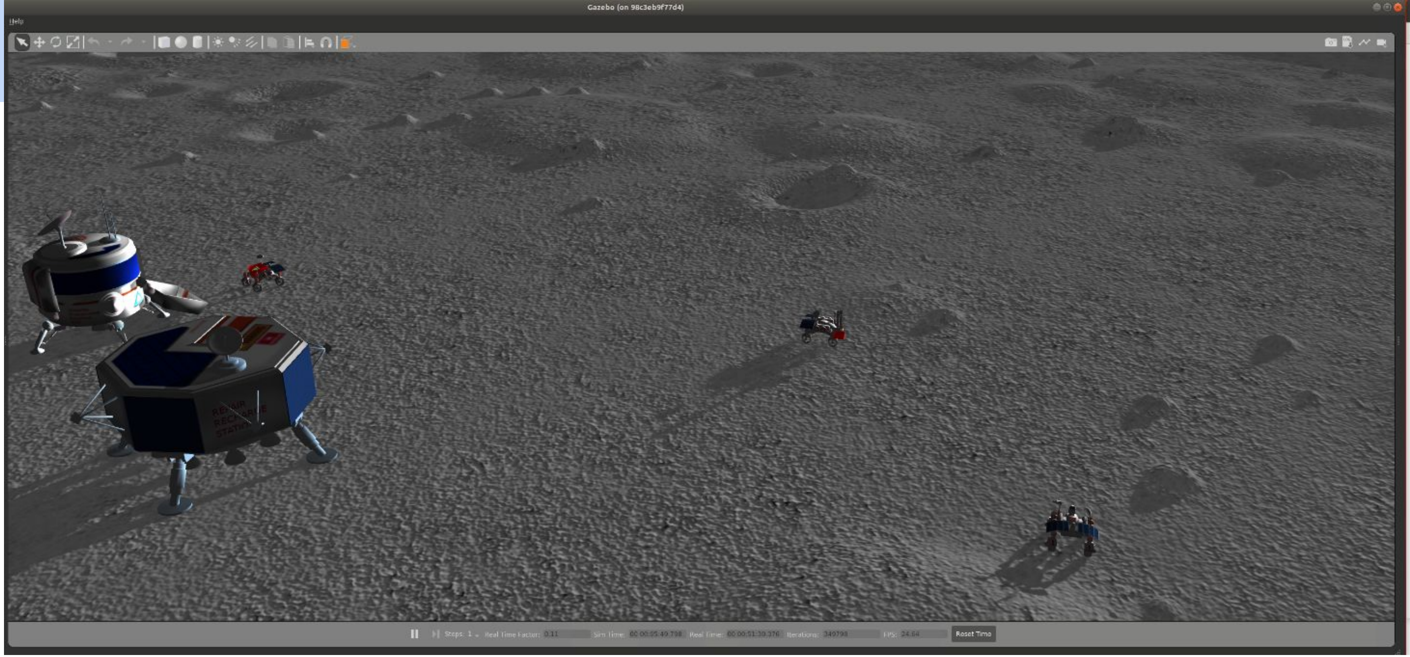
2 large landers (Processing Plant + Repair Station)

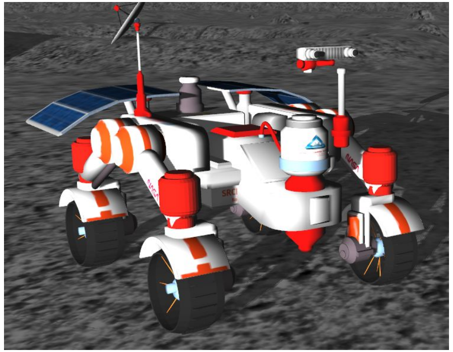
Random rocks / craters / hills / volatile rich regions

Goal:

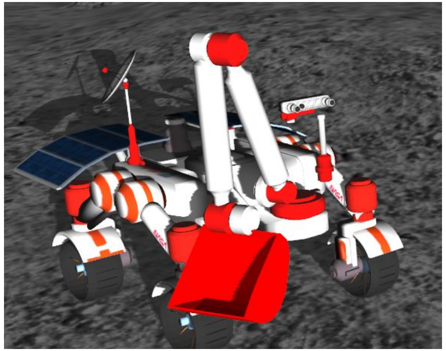
- discover volatile rich areas
- excavate the regolith of these regions
- haul it back to the Processing Plant

Three rover types ->

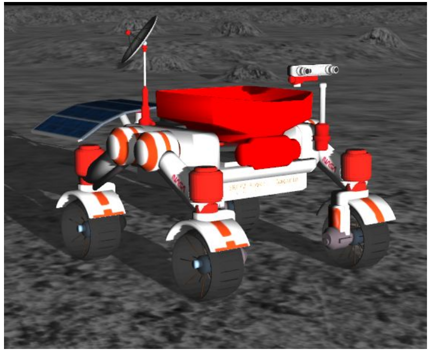




Scout



Excavator

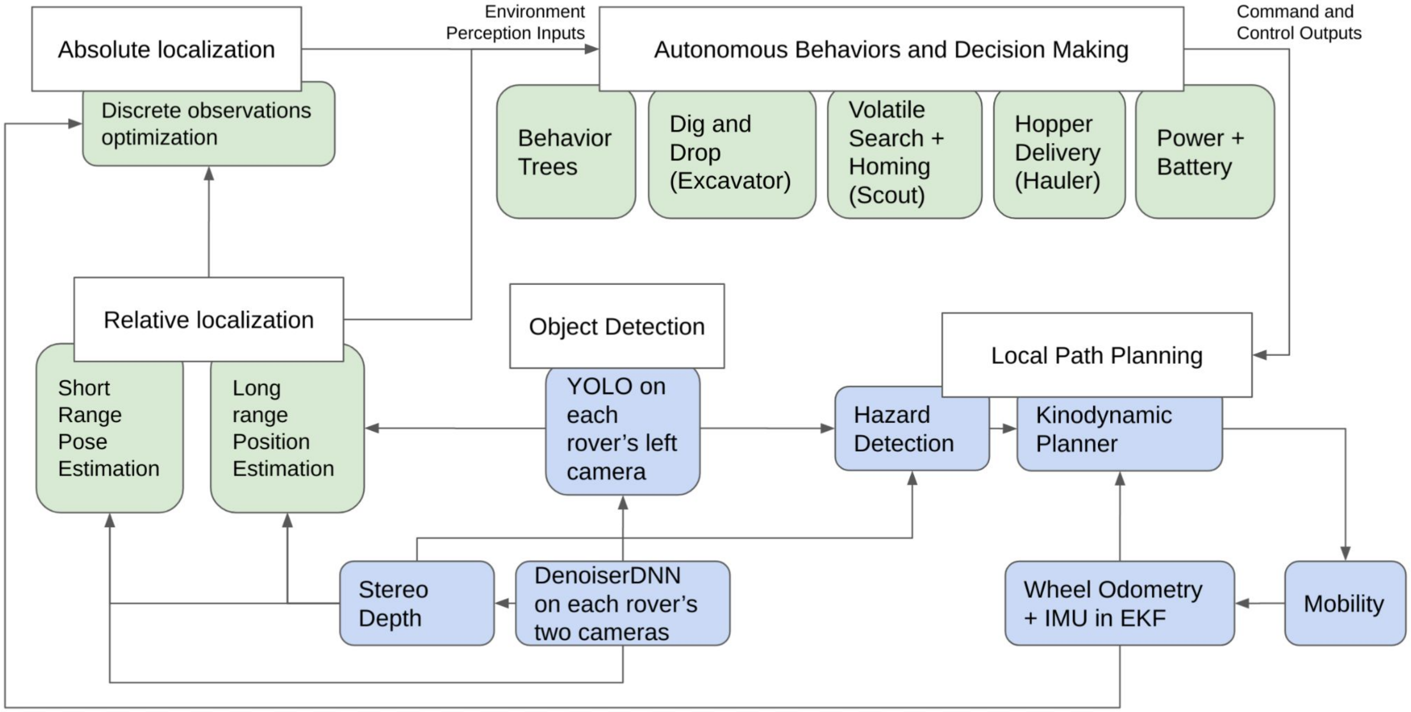


Hauler

Concept of Operations and System Architecture

Robotic team:
- 1 scout
- 1 excavator
- 1 hauler

Sequential strategy to
- reduce absolute localization requirements
- scoring cycles of 20 minutes
  -> repeat for 2 hours



Blue: Infrastructure: low-level functions and those performed for each rover individually
Green: Collaboration: high-level functions and those that require collaboration between 2 or more rovers

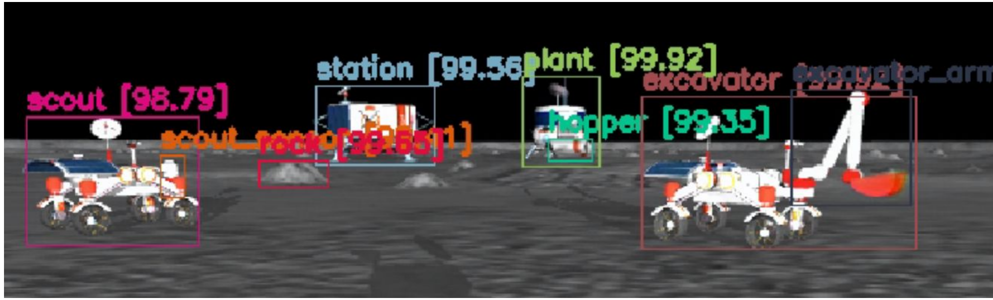
Focus

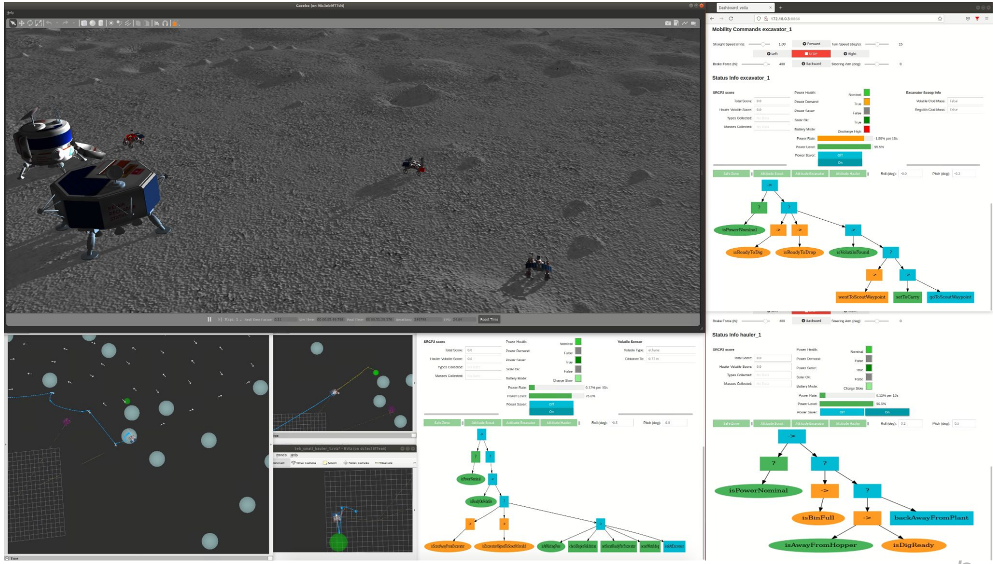
Ideas for use in future NASA / lunar surface robotics missions

1. Integration of deep learning methods into the robotics stack

2. Autonomous multi-rover collaboration with Behavior Trees

3. Visualization and development methodology for autonomous robotics





1. Integration of deep learning methods

Image De-Noising via Deep Neural Network [1]

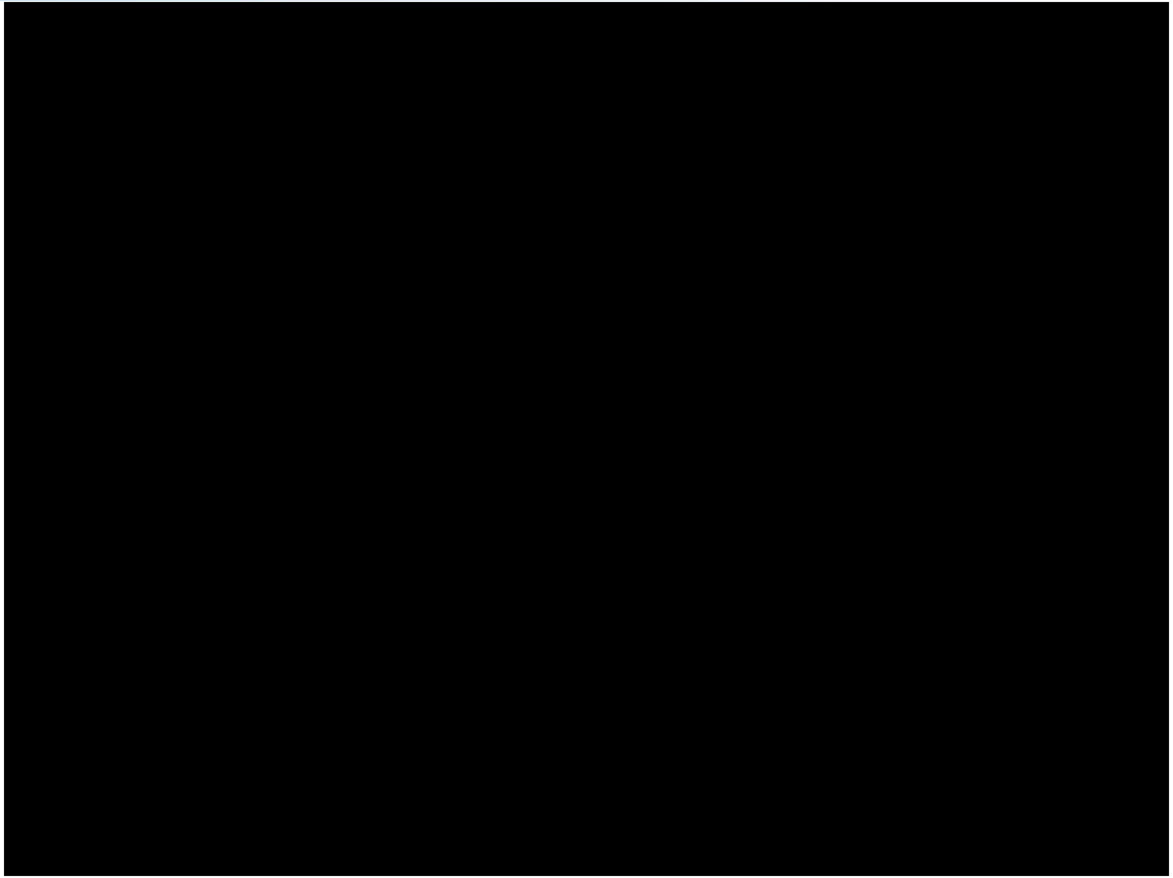
Top : raw, noisy images

- Edges corrupted
- Gaussian + salt and pepper noise

Bottom: de-noised images

Left: stereo point cloud computation

6 cameras de-noised +
3 stereo point clouds computed in real time at 2 Hz



[1] Tassano, M, Delon, J and Veit, T 2020 ‘Fastdvdnet: Towards real-time deep video denoising without flow estimation’, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 1354-1363).

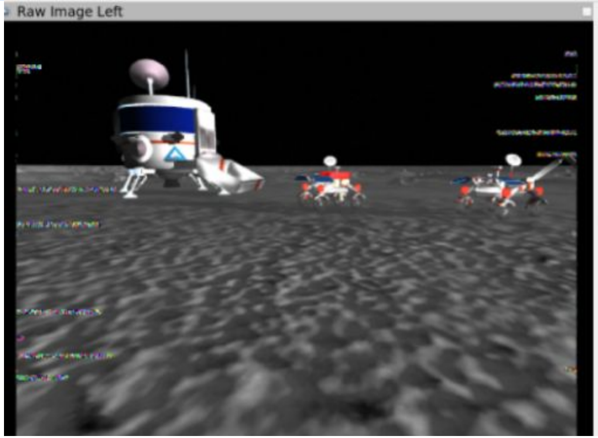
1. Integration of deep learning methods

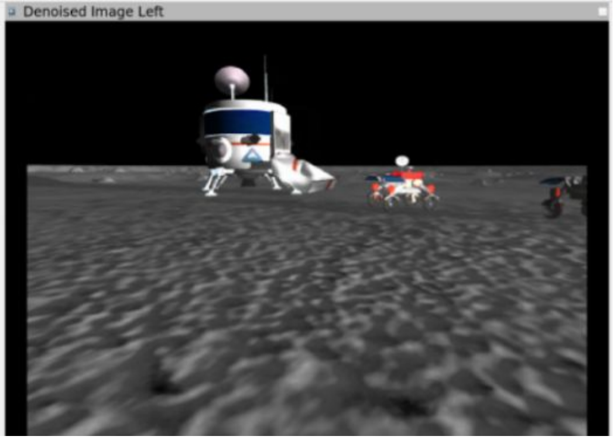
Image De-Noising via Deep Neural Network

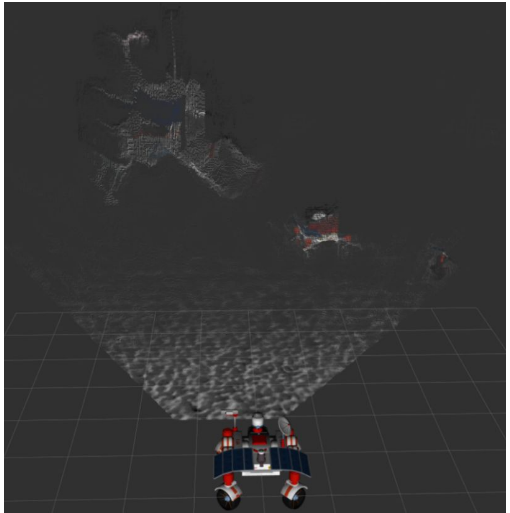
Top : raw, noisy images
- Edges corrupted
- Gaussian + salt and pepper noise

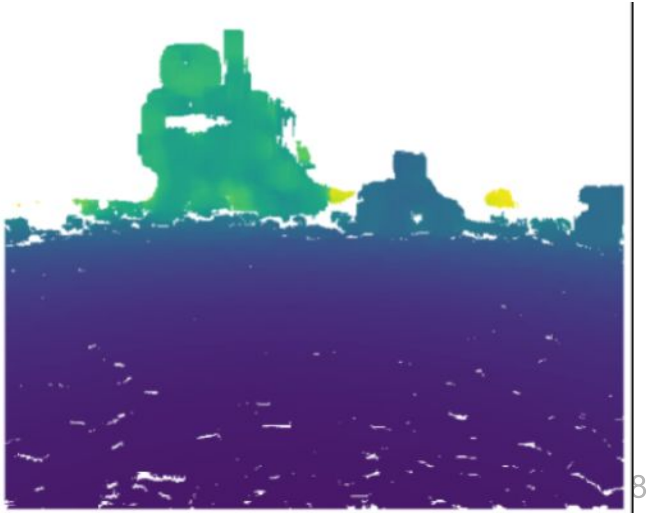
Bottom: de-noised images
Left: stereo point cloud computation

6 cameras de-noised +
3 stereo point clouds computed in real time at 2 Hz









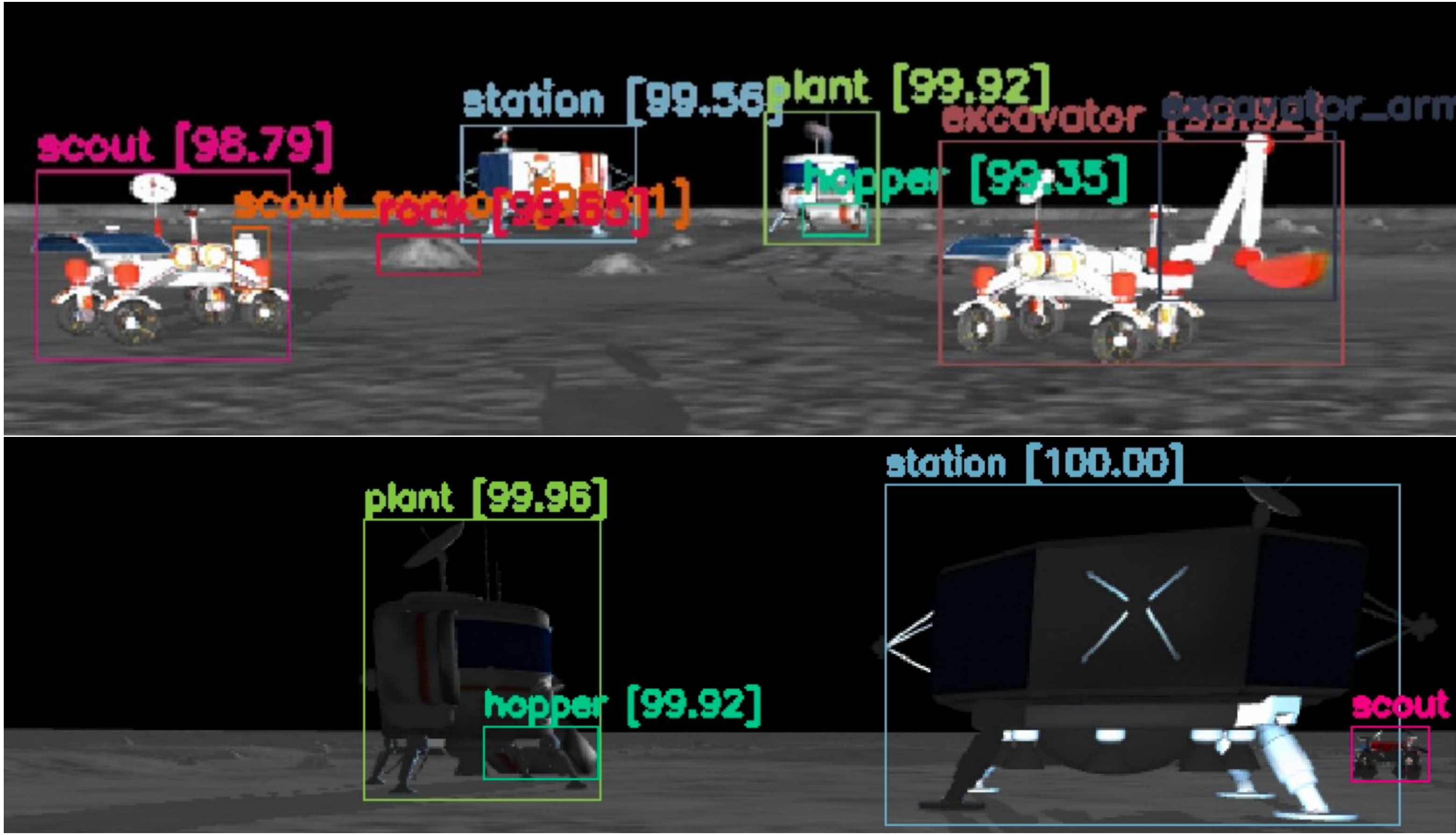
1. Integration of deep learning methods

YOLO Object Detection [2]

15 classes for environment awareness and Hazard Detection

Performed on each rover's camera at 2 Hz

Robust to lighting and orientation conditions



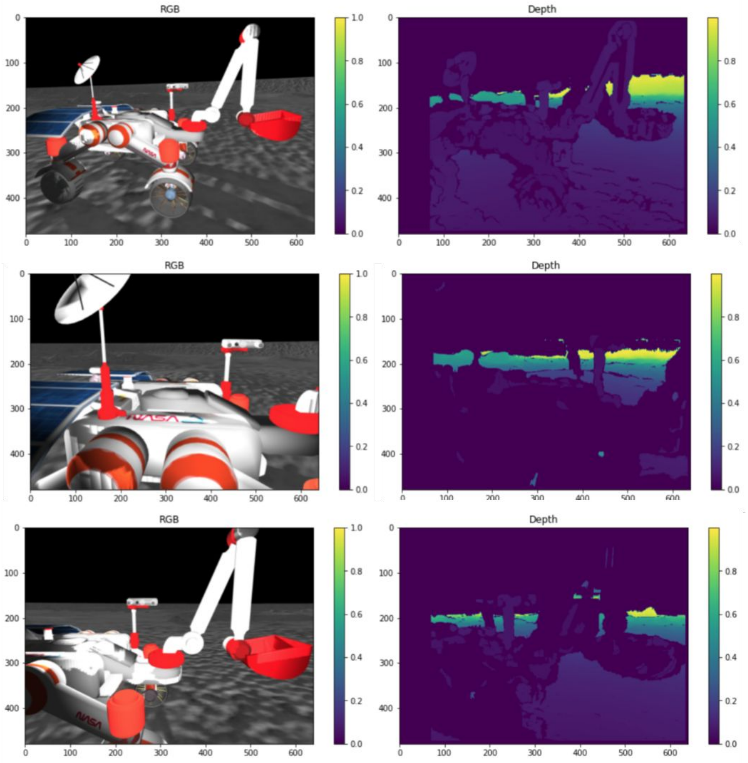
[2] Redmon, J, Divvala, S, Girshick, R and Farhadi, A 2016 'You only look once: Unified, real-time object detection', *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).

Relative Pose estimation

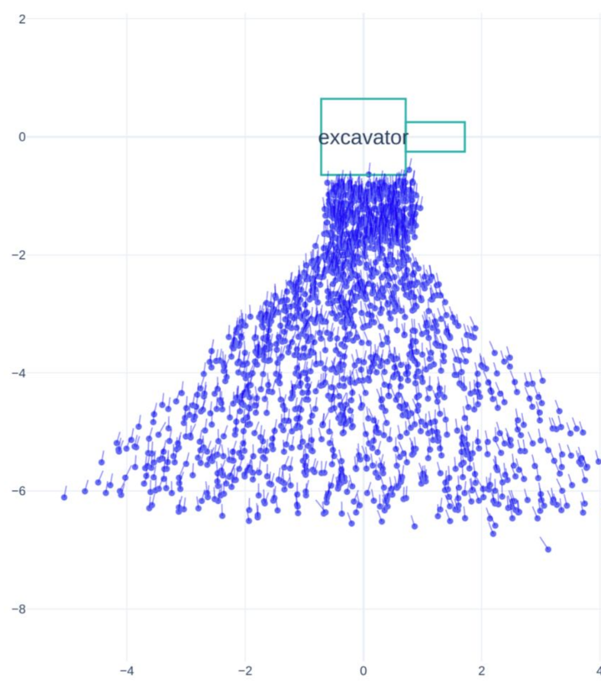
For rover alignements

Training data: ~10k images of rovers next to each other + gazebo ground truth relative pose

Inference : Input single camera image -> output relative position and orientation



Samples collected:
Train = 1680 / Validation = 480 / Test = 240 / Total = 2400 samples



2. Autonomous multi-rover collaboration with behavior trees

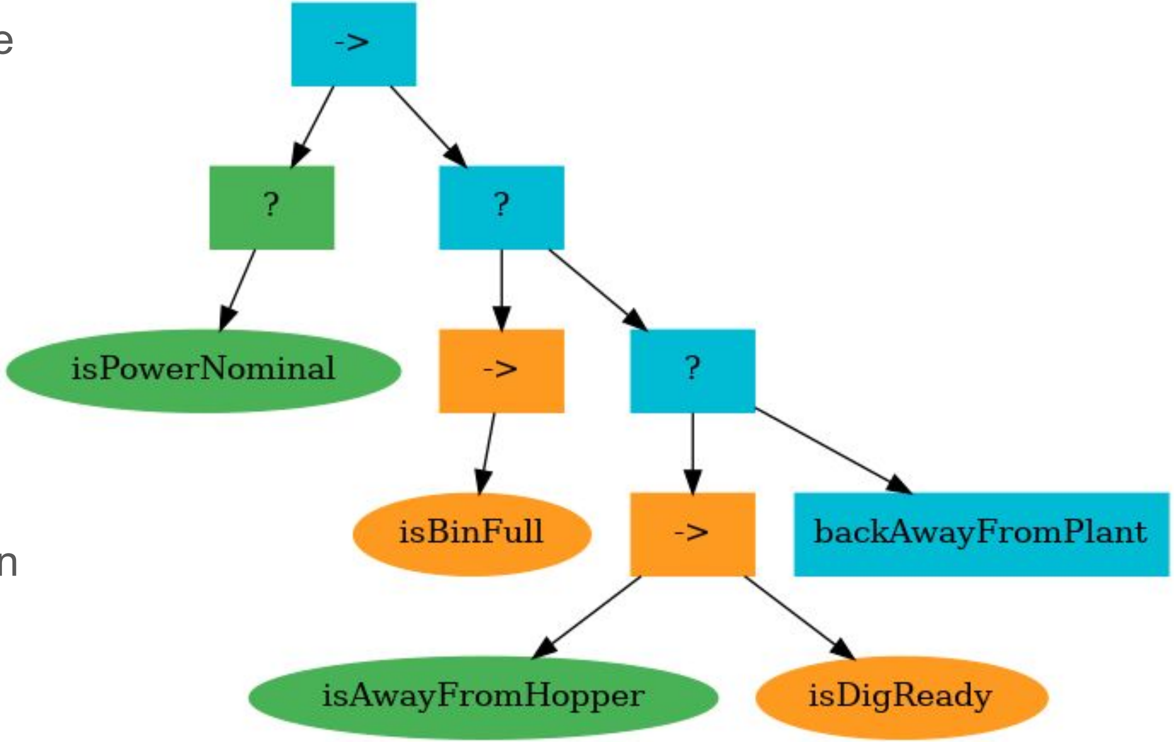
More modular and reactive than state machines

Developed our own implementation for simplicity and visualization

Can design rover behavior from simple to complex and recovery

Natural conditions for synchronisation between multiple rovers

Can follow what each rover is “thinking” during development and testing



running, success, failure

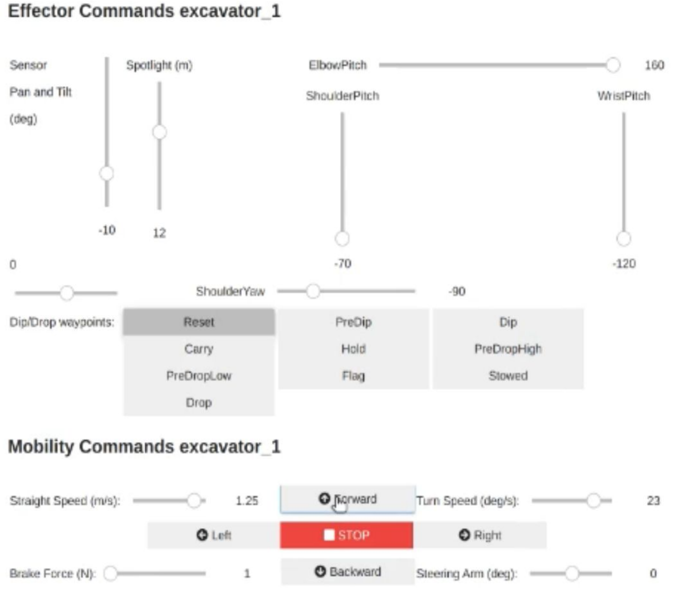
3. Visualization for autonomous robotics

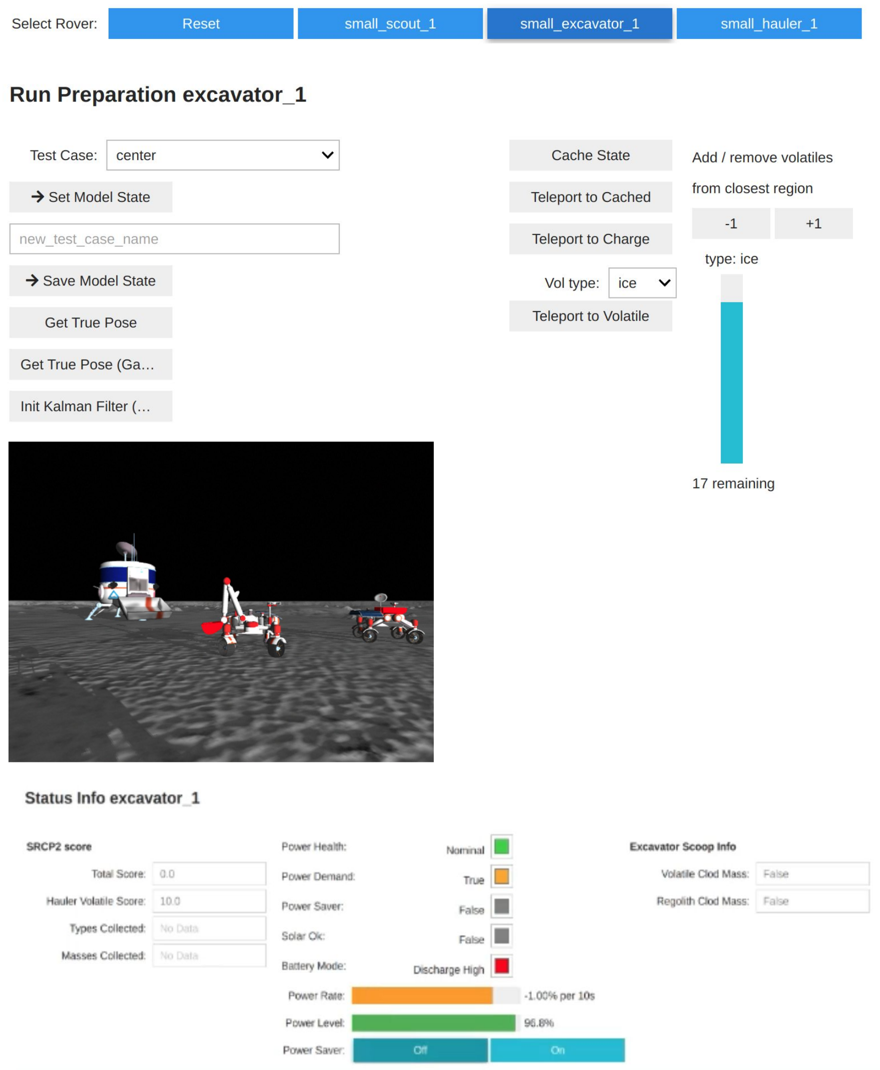
Development:

Never use the command line interface to control rovers or watch ROS topics

Our own Dashboard interface

-> Low friction + intuitive development



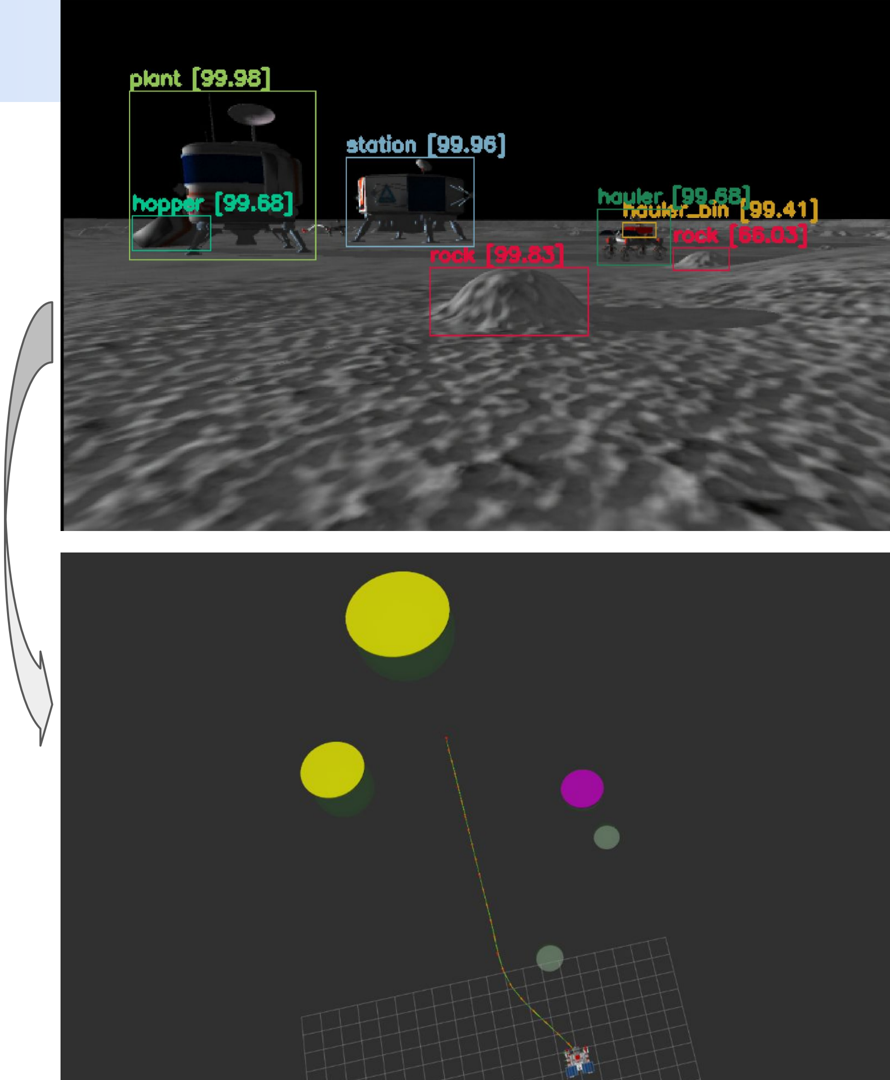


3. Visualization for autonomous robotics

Visualization of outputs of algorithms and modules

- YOLO + stereo point cloud + discretization of objects (colored disks)

- Current local goal and trajectory planning around hazards (green path)



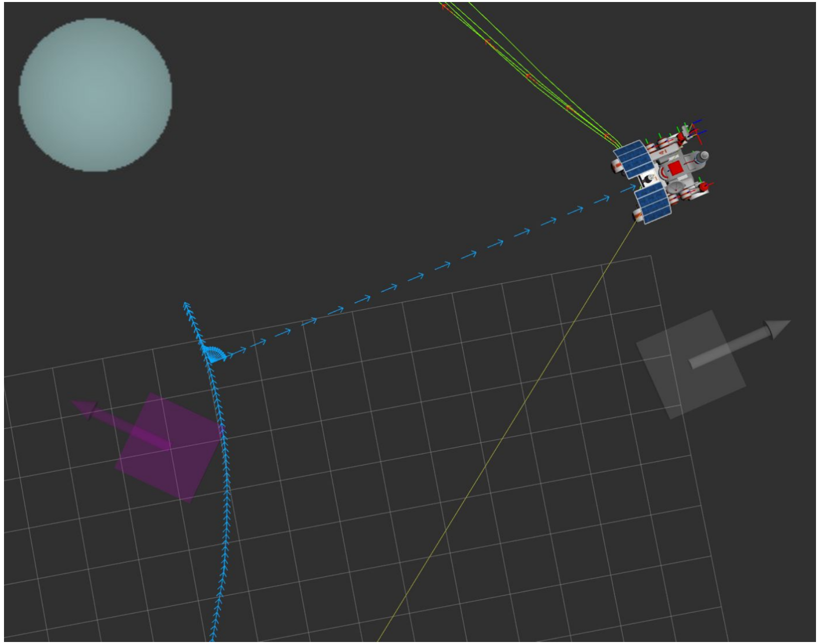
3. Visualization for autonomous robotics

Visualization of outputs of algorithms and modules

- Rover past trajectory from odometry (blue path)
- Rover current position from localization (rover mesh)
- Ground truth from Gazebo
  - Grey box for current rover
  - Purple box for other rovers
  - Blue disk for volatile regions

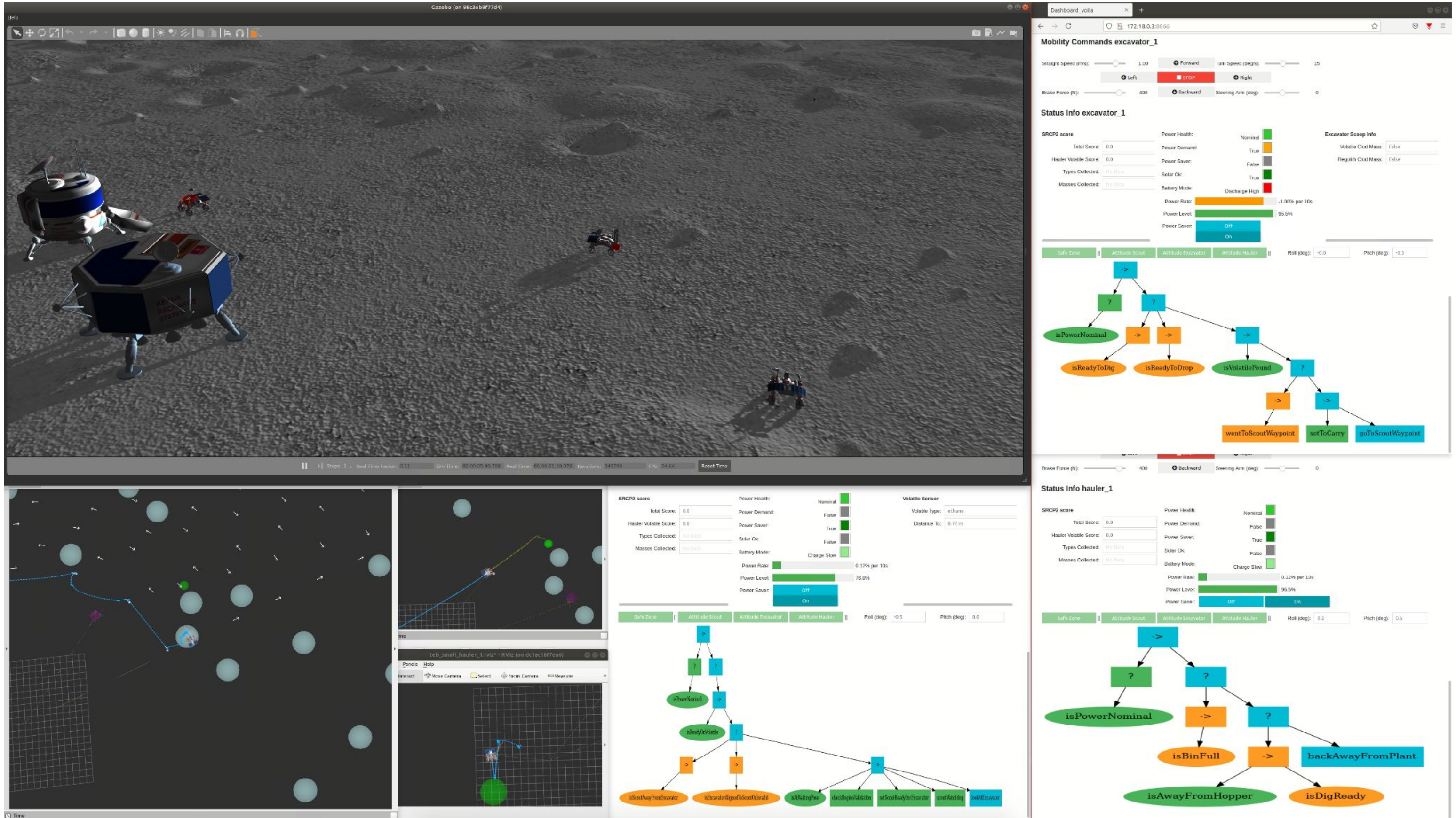
Odometry + Localization

Ground truth from Gazebo

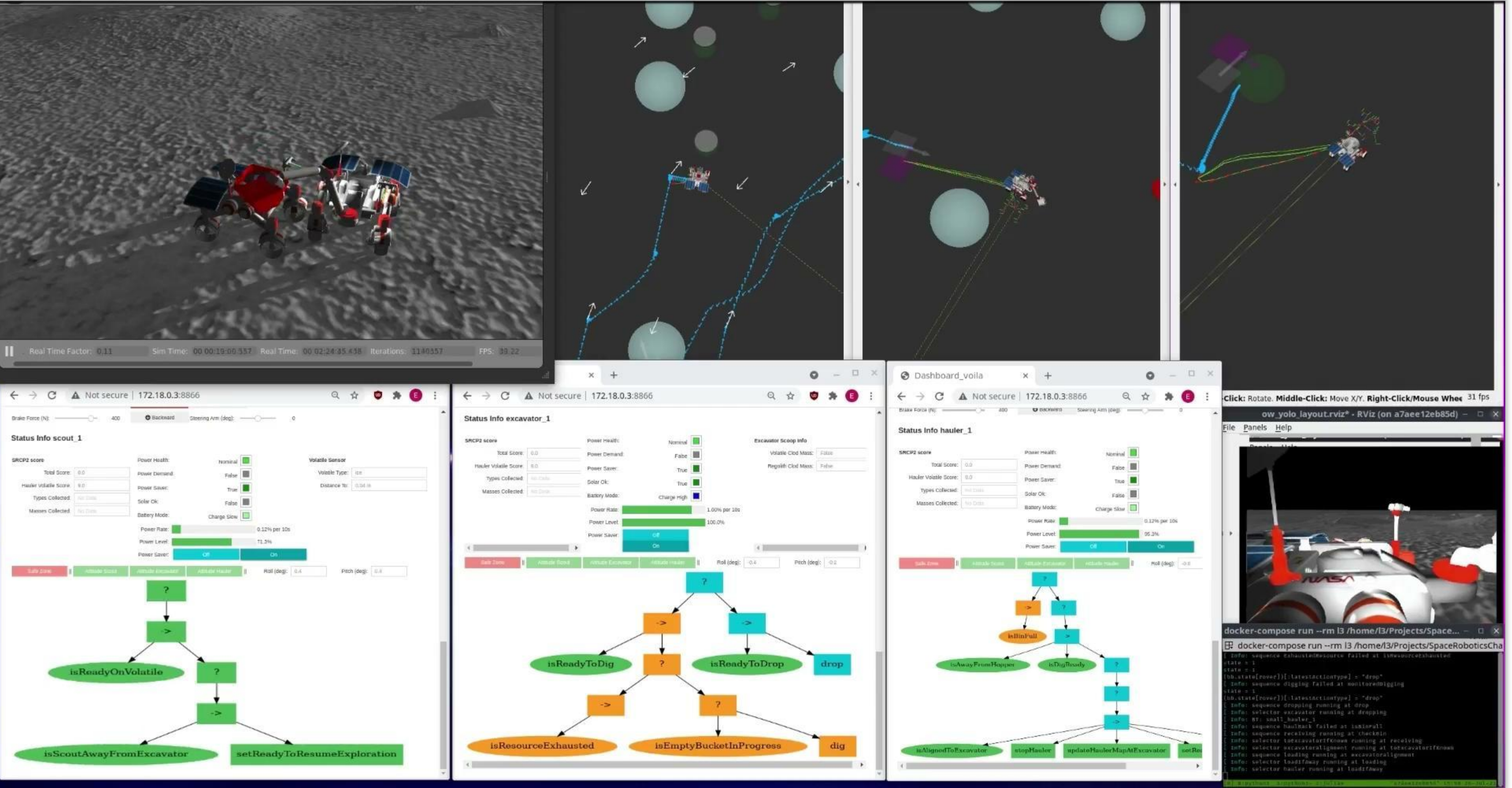


3. Visualization for autonomous robotics

Integrated tests with three rovers -> deep insight into issues and root cause -> faster fixes



8. Excavator digs and drops volatiles into Hauler's bin



Thank you

To the developers, engineers, project managers at NASA and NineSigma who made the Space Robotics Challenge possible

To the other teams for challenging each other and helping making the simulation better via gitlab issue tracker

To our collaborators OffWorld Inc. and Mission Control Space Services Inc., who contributed resources and advice + invaluable team members

To the developers of the many open-source tools and the ROS/Gazebo communities

We open sourced the behavior tree library -> https://github.com/fabid/BehaviorTree.jl

We released the video on youtube ->

We will release the Electronic Summary soon!