URSSA: A Simulator for Lunar Surface Telerobotics Research

M. S. Menon, M. Walker, D. Koris, D. Szafir, J. Burns
University of Colorado, Boulder

Motivation

As extra-planetary terrain exploration missions complex & robotics intensive, we should adapt autonomous algorithms (efficient use of operator time) & extensive mission mockups (makes system design robust). Over the years, mission analogs have been the best bet for mission mockups and algorithm testing.

However, difference in governing physical processes for topography and photometry, and hurdles of logistics/ time overhead make mission analogs increasingly difficult for such missions.

CU Boulder NESS Team Proposal

- Explore prospect of using virtual environment simulators to test visual-inertial navigation algorithms & do mission mockups (with Head-Mounted Displays).
- Develop URSSA - Unity™- ROS™ (Robotic Operating System) Simulator for Space Applications. Unity simulates the environment & ROS simulates the robot.
- Do an application case-study in Lunar environment.

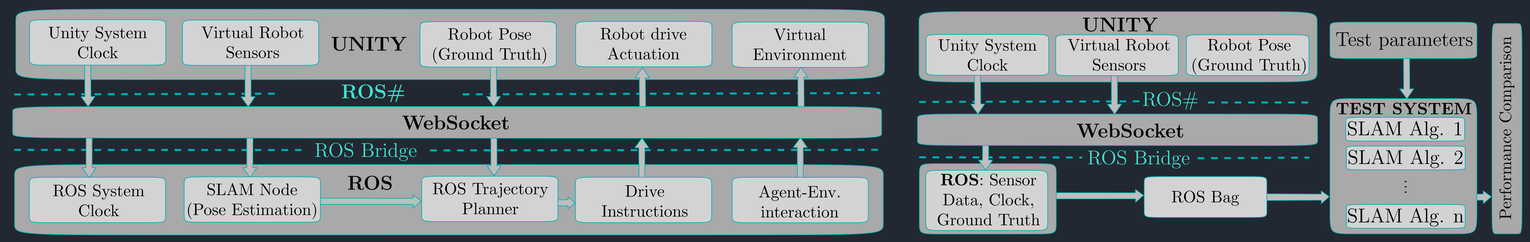
What is Unity™, ROS™?

Unity is a cross-platform game engine with 3D physics simulator. **ROS** software framework provides services for hardware abstraction & simulation of robots.

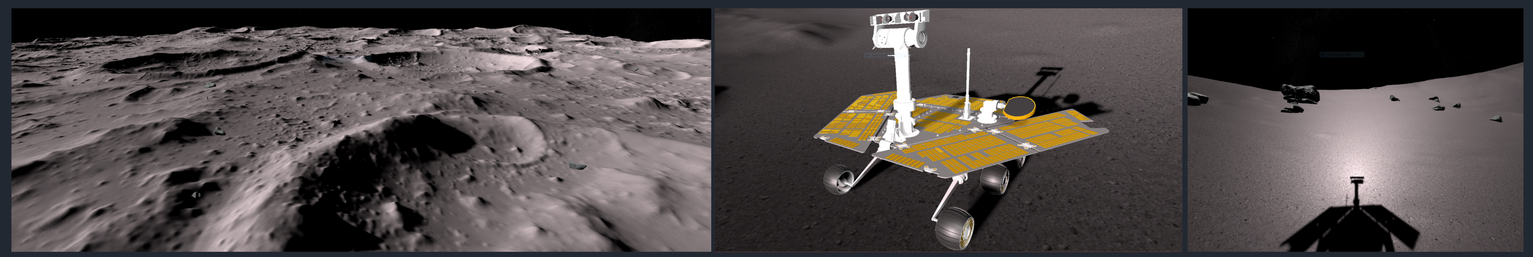
Simulator Features Implemented

- Lunar Photometry (Opposition Surge, no limb darkening etc.) is modeled by custom Hapke BRDF (Bidirectional Reflection Distribution Function) shader.
- Terrain from LRO-NAC DEMs with fractal expansion
- Tire tracks, agent-environment interaction modeled
- Dynamic Tessellation for memory efficiency
- Small craters, boulders placed on terrain using available Size-Frequency-Distributions
- IMU, LiDAR & visual camera sensor models
- Wheel-terrain Coulomb friction modeled
- ROS# framework for ROS-Unity communication
- Time synchronization between ROS & Unity
- Frame rate is 8 fps without any code optimization

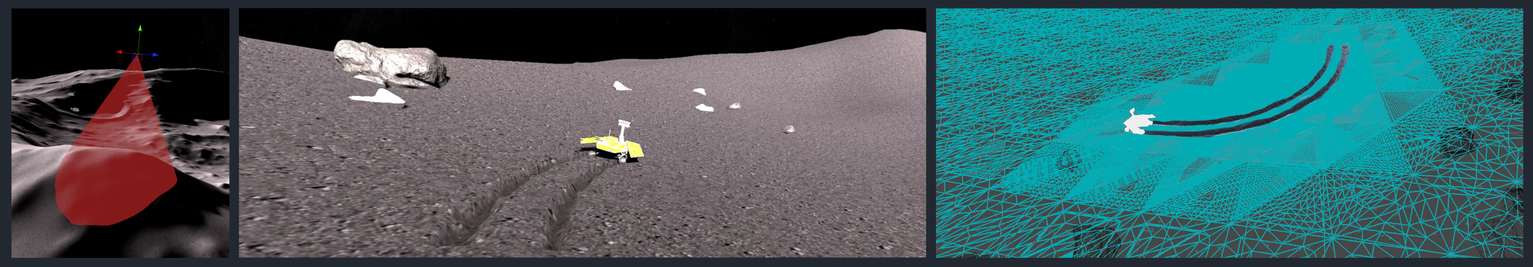
Extra-terrestrial Environment Simulators help researchers to run **Virtual Mission Analogs** to **Test On-board Algorithms** & **Do Mission Mockups**. We can realize **Fast, Modular** & **Scalable** architectures for these simulators using **Game Engines.**



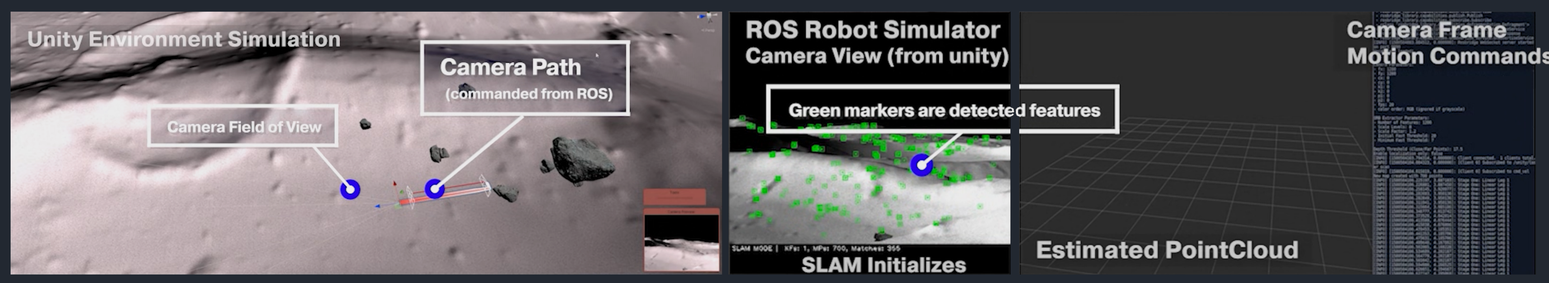
Simulator Architecture



Lunar terrain**(L)** & Rover model**(C)** used in the case-study, Rover view showing Opposition Surge**(R)**



Virtual LiDAR model**(L)**, Tire tracks & synthetic boulders**(C)**, Dynamic Tessellation by wheel proximity**(R)**



Testing ORB-SLAM2 (Simultaneous Localization And Mapping) algorithm on URSSA

We are open to collaborations to develop & release the simulator open source. If interested, contact midhun.menon@colorado.edu
This work is directly supported by the NASA-SSERVI cooperative agreement 80ARC017M0006.

NESS Network for Exploration and Space Science