URSSA: A SIMULATOR FOR LUNAR SURFACE TELEROBOTICS RESEARCH. Midhun S. Menon[1], Michael E. Walker[2], Dan Koris[2], Daniel Szafir[2,3], and Jack Burns[1], [1]Center for Astrophysics and Space Astronomy, University of Colorado, Boulder (Email:midhun.sreekumar@gmail.com), [2]Department of Computer Science, University of Colorado, Boulder, [3]ATLAS Institute, University of Colorado, Boulder.

Introduction: Future space missions and activities will rely on robotic systems. Primary reason for this is the safety concern associated with sending human astronauts to such hostile and uncertain environments. Add to this the fact that future missions will encompass building sustainable infrastructure on extraterrestrial bodies, which will call for advanced surface telerobotics and robotic assembly tasks. However, designing such robotic systems for space exploration is still an open challenge.

Designers rely on field analogs to test system performance. However, in this scenario, field analogs have limited scope because of inability to simulate all the critical and unique aspects of the target environments like topography, photometry, hardware/ logistics barriers, etc. On top of this, getting large enough training data-sets for training such robotic systems will be another hard task. Hence, development of virtual planetary environment simulators is important for preparing for future missions in terms of algorithm design/prototyping, operations planning, autonomous agent training and mission mock-ups.

We propose URSSA (Unity-ROS Simulator for Space Applications), a scalable and modular simulation framework which generates physically accurate lunar virtual environment. Our framework uses the Unity game engine to simulate the environment and Robotic Operating System (ROS) to simulate the autonomous agents/ robots operating in the environment. The asynchronous bilateral communication between ROS and Unity is implemented with ROS#.

Implementation:

Environment (Unity). We simulate the lunar topography with high resolution terrain modeled by overlaying synthetic roughness (by fractal expansion using derived statistical measures of the terrain roughness from Apollo missions) on the Digital Terrain Models generated from Lunar Reconnaissance Orbiter-Narrow Angle Camera. Lunar photometric effects like Opposition Surge is modeled using terrain shaders based on Hapke Bidirectional Reflectance Distribution Functions (BRDF). Tire tracks have been modeled with wheel proximity based dynamic tessellation for memory efficiency. Small boulders and craters are probabilistically distributed using observed Size-Frequency-Distributions of lunar terrain.

Robotic Systems (ROS). We model the robots in ROS. They take inputs from the virtual environment in Unity via multiple virtual sensors like mono/ stereoscopic cameras, LIDAR, Inertial Measurement Unit (IMU), odometers, etc. The robots carry out path and motion-planing for resource prospecting, tele-operation and assembly tasks.

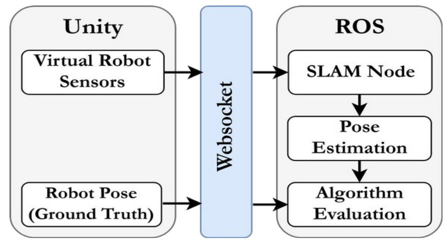

Figure 1: Simulator Architecture

Experiment and Discussion: We tested mono and stereo versions of ORBSLAM2, a class of navigation algorithm on the simulator. We made the rover to traverse a known straight path on the terrain of predetermined length at constant velocity. During the simulation, ROS receives time stamped sensor outputs (Cameras, IMU etc.). We compare the ORBSLAM2 trajectory estimate with the ground truth data to evaluate performance of the SLAM algorithm. As expected, we found stereo cameras to do better.

Another observation was the delay in initialization/ loss of tracking for ORBSLAM2 for the dearth of features/ low image gradients (barren flat terrain, lack of boulders/ craters) and heavy opposition surge/ glare on lunar landscape (adversarial Hapke coefficients).

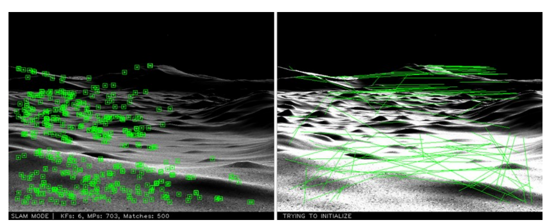

Figure 2: Loss of SLAM features/ tracking under adversarial photometric conditions

Conclusion and future work:

The results show the potential of using the simulator for testing and design of navigation algorithm. We intend to do a more extensive comparison among multiple SLAM algorithms to analyze performance and create featureless SLAM algorithm for navigation in lunar environments.

Acknowledgments: This work is directly supported by the NASA Solar System Exploration Virtual Institute cooperative agreement 80ARC017M0006.