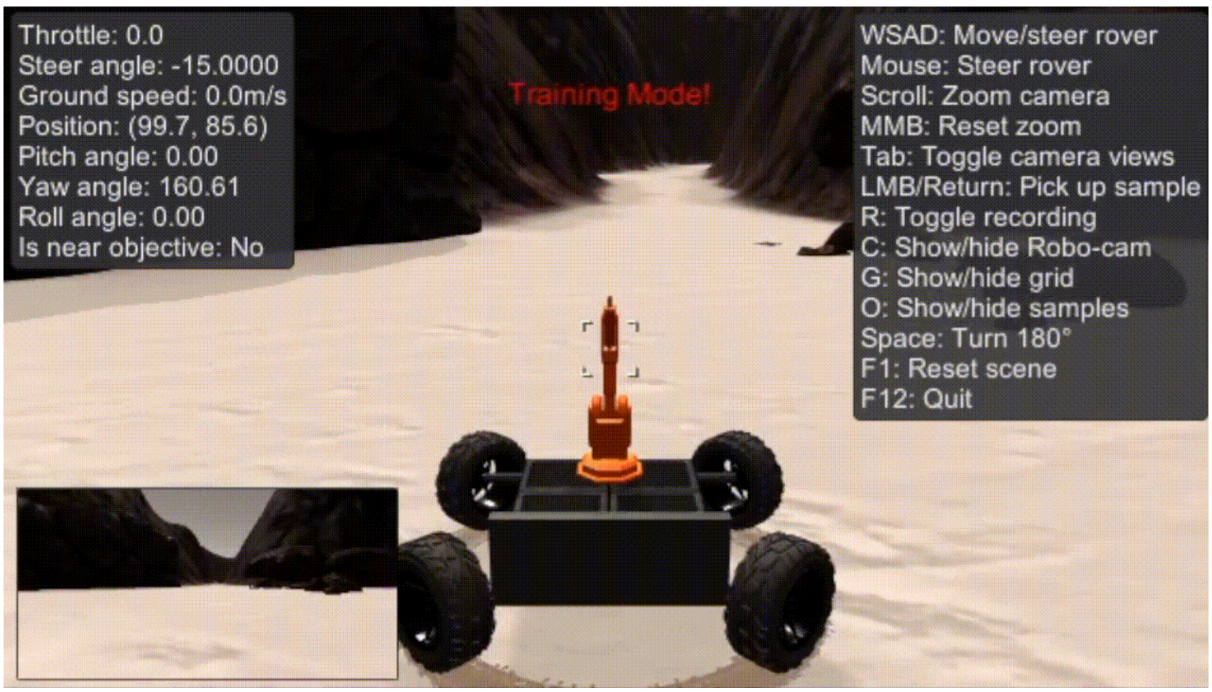
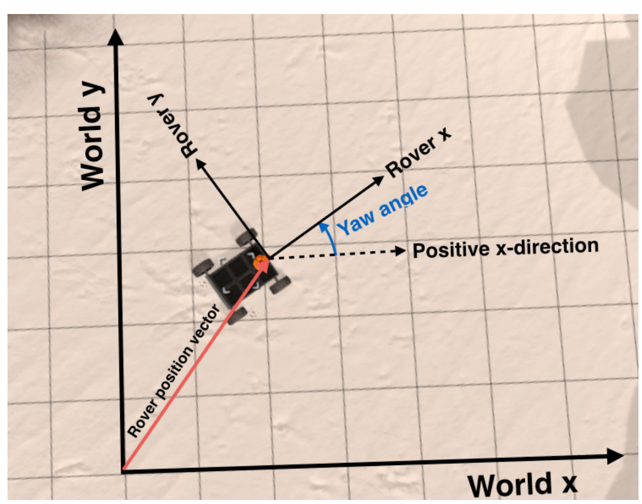
NASA Mars Sample & Return Rover (MARS MSR)

For at least three decades, scientists have advocated the return of geological samples from Mars. One early concept was the Sample Collection for Investigation of Mars (SCIM) proposal, which involved sending a spacecraft in a grazing pass through Mars's upper atmosphere to collect dust and air samples without landing or orbiting.

As of late 1999, the MSR mission was anticipated to be launched from Earth in 2003 and 2005. Each was to deliver a rover and a Mars ascent vehicle, and a French supplied Mars orbiter with Earth return capability was to be included in 2005. Sample containers orbited by both MAVs were to reach Earth in 2008. This mission concept, considered by NASA's Mars Exploration Program to return samples by 2008, was cancelled following a program review



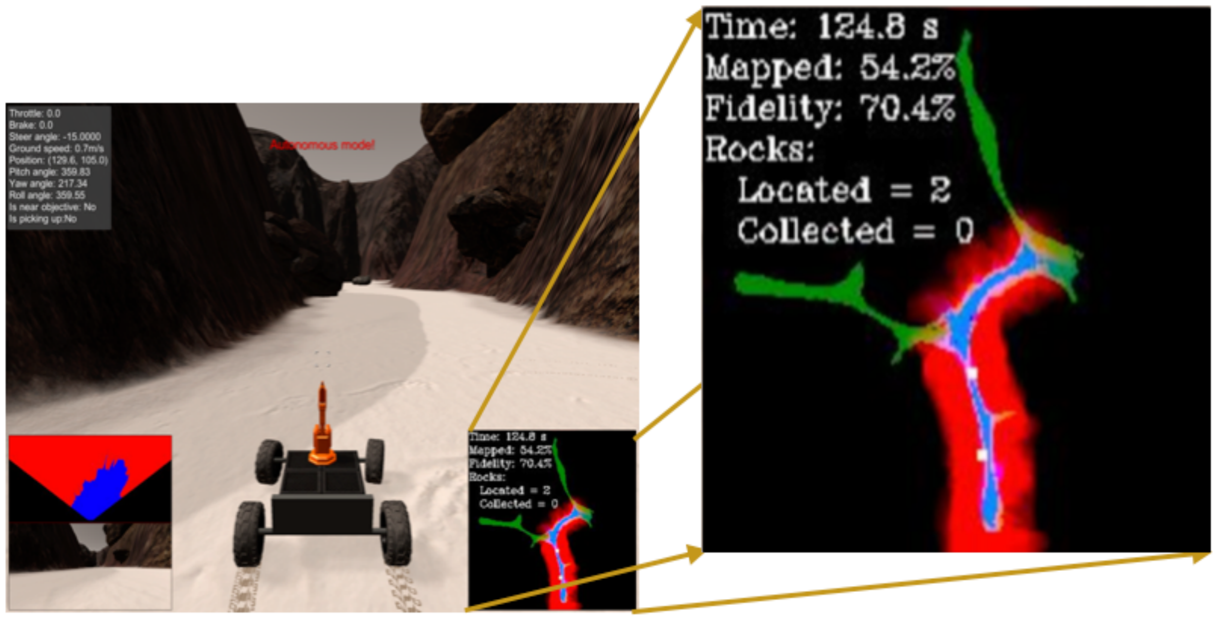
In this project, we'll do computer vision for robotics. We are going to build a Sample & Return Rover in simulation. Mainly, we'll control the robot from images streamed from a camera mounted on the robot. The project aims to do autonomous mapping and navigation given an initial map of the environment. Realistically speaking, the hard work is done now that you have the mapping component! You will have the option to choose whether to send orders like the throttle, brake, and steering with each new image the rover's camera produces.



Phase 1 – Basic Operation:

The inset image at the bottom right when you're running in autonomous mode is packed with information. In this image, your map of navigable terrain, obstacles and rock sample locations is overplotted with the ground truth map. In addition, some overall statistics are presented including total time, percent of the world you have mapped, the fidelity (accuracy) of your map, and the number of rocks you have located (mapped) and how many you have collected.

The requirement for a passing submission of the first phase is to map at least 40% of the



environment at 60% fidelity and locate at least one of the rock samples (note: you're not required to collect any rocks, just map the location of at least 1). Each time you launch the simulator in autonomous mode there will be 6 rock samples scattered randomly about the environment and your rover will start at random orientation in the middle of the map.

.

Expected output:

For the first phase, the expected output is as follows:

- Your pipeline should be able to map at least 40% of the environment at 60% fidelity. It should repaint the map image to distinguish between navigable terrain, obstacles and rock samples.
- You're required to locate at least one rock in the environment.
- You should also implement debugging mode where each step of your pipeline is illustrated with the vehicle operation.

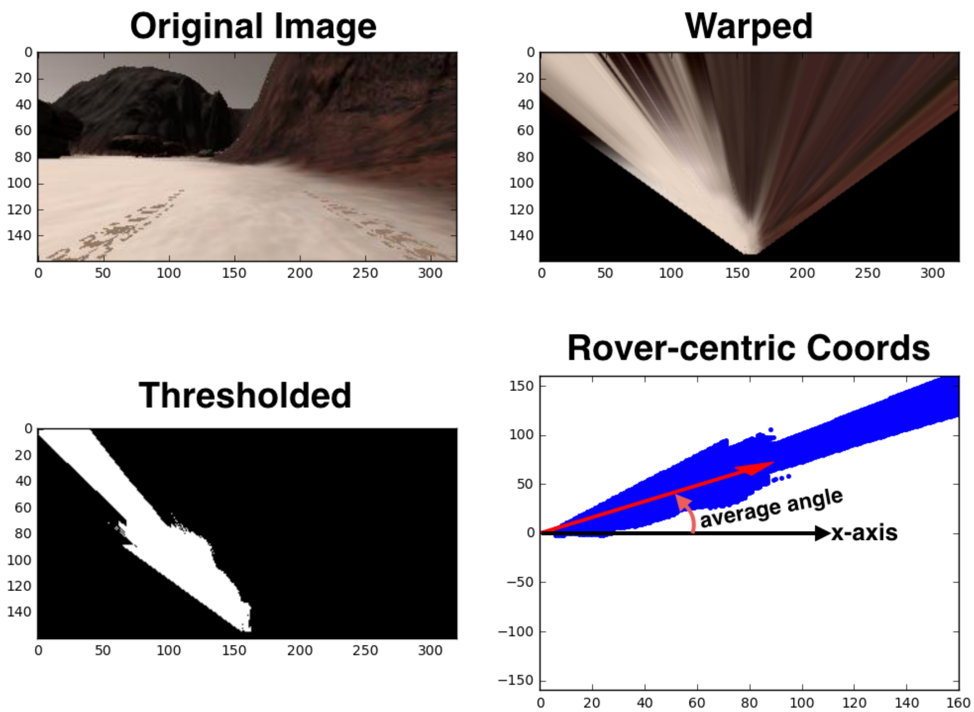
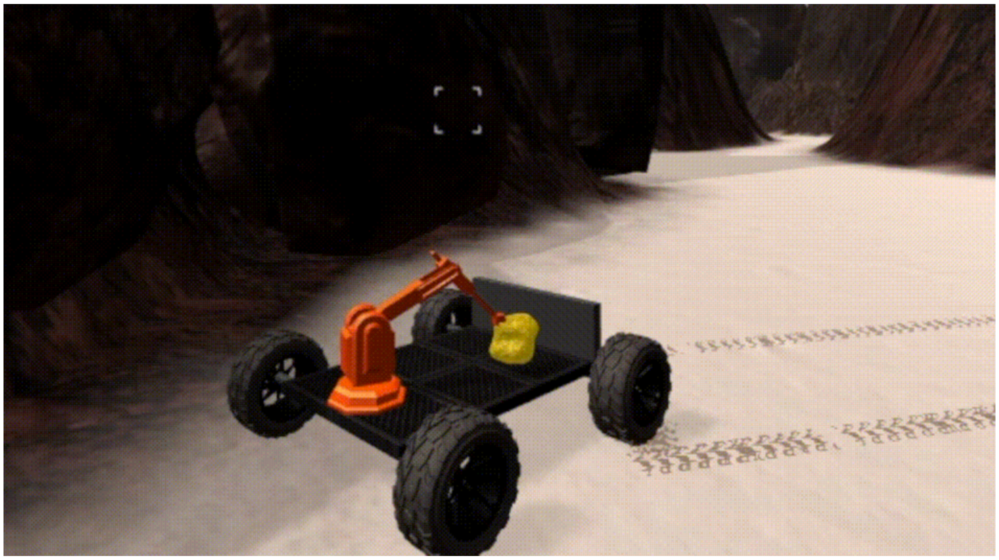


Figure 1 In this example, the different stages of the pipeline are shown.

Phase two – Collect & Return

In this stage, build upon your previous implementation to map at least 95% of the environment at 85% fidelity. All while colliding with the least number of obstacles. (The maximum number of collisions allowed will be announced at the beginning of phase 2)

Also, there is a robotic arm located on the vehicle. In this phase, you should also locate and use the robotics arm to pick up at least five rocks out of the six, and then return them back to the start position.



Optimizing Metrics: The primary metrics of interest are time, percentage mapped, fidelity and number of rocks found. In the best-case scenario, you would map the entire environment at a very high fidelity and locate and collect all six rock samples in the minimum total amount of time. To do this you'll need to not only optimize the accuracy your mapping analysis, but also the efficiency with which you traverse the environment.

Optimizing Map Fidelity Tip: Your perspective transform is technically only valid when roll and pitch angles are near zero. If you're slamming on the brakes or turning hard, they can depart significantly from zero, and your transformed image will no longer be a valid map. Think about setting thresholds near zero in roll and pitch to determine which transformed images are valid for mapping.

Optimizing Time Tip: Moving faster and more efficiently will minimize total time. Think about allowing for a higher maximum velocity and give your rover the brains to not revisit previously mapped areas.

Optimizing % Mapped Tip: Think about ways you can "close" boundaries in your map. If you can do this effectively, then once you close all boundaries your map will be complete!

Optimizing for Finding All Rocks Tip: The rocks always appear near the walls. Think about making your rover a "wall crawler" that explores the environment by always keeping a wall on its left or right. If done correctly, this optimization can help all the metrics.

Simulator and Starter Code.

You will be provided with a simulator (running on Linux), and a starter code. You should build upon them to complete the project.

It's IMPORTANT to note that the project will ONLY BE ACCEPTED WHEN RUNNING ON LINUX. NO OTHER OPERATING SYSTEMS ARE ALLOWED AND NO EXCEPTIONS WILL BE MADE

What to be delivered for each phase:

- The code well written and commented.
- A Github repository with your code on it. The Github repository should be your main tool for different stages of the project (You shouldn't only push the code when you're done with each phase). The readme should also have clear instructions on how to run your code
- A Jupiter-notebook showing the result of your pipeline on the provided test images.
- The output of your pipeline on the provided test videos.
- A report containing the methods used in the pipeline and explaining them in detail
- Finally, your code should support a debugging mode whether it was a video or a single image. When this mode is activated, your pipeline should be showing all the stages that your code is going through.

Teams' formation:

You're encouraged to work in teams. The maximum of each team is 5.

Deadline:

Phase one:

December 5th, 2022, 11:59 pm

Phase two:

December 31th, 2022, 11:59 pm

The late policy is as follows: for each late hour of the first four hours, 20% of the total mark will be deducted. After the 4 first hours, only 20% of the total marks will be given without any further deductions. A late hour can be anything from one second to sixty minutes. No exceptions will be made.

Good luck