Automated lander simulation utilizing ZEM ZEV guidance with optical rotation estimation

Ashleigh Sinclair & Dr Kit Ying Hui

Introduction

There are over 1 million asteroids in our solar system. Each of potential scientific interest and comprised of valuable resources. Varying in size from 530km across to just 10m. All with differing shapes and rotational velocities. Given this high variation between objects combined with the difficulties involved in remote landing operations across vast distances, such as a sizeable communications delay, a lander capable of autonomous navigation would be necessary to explore these bodies fully.

Project Aim

Develop a simulation allowing for experimentation of automated landing techniques combined with environment sensing capabilities which can then be used as a test platform to gauge performance of flight characteristics, automation strategies, and investigate how environment variables impact sensing equipment and performance.

Framework

The framework for the simulation was written in C++. The Bullet library was used for flight and collision calculations. Rendering of the environment is performed with the Vulkan API, with an off screen rendering pass used to simulate the landers onboard camera.

Method

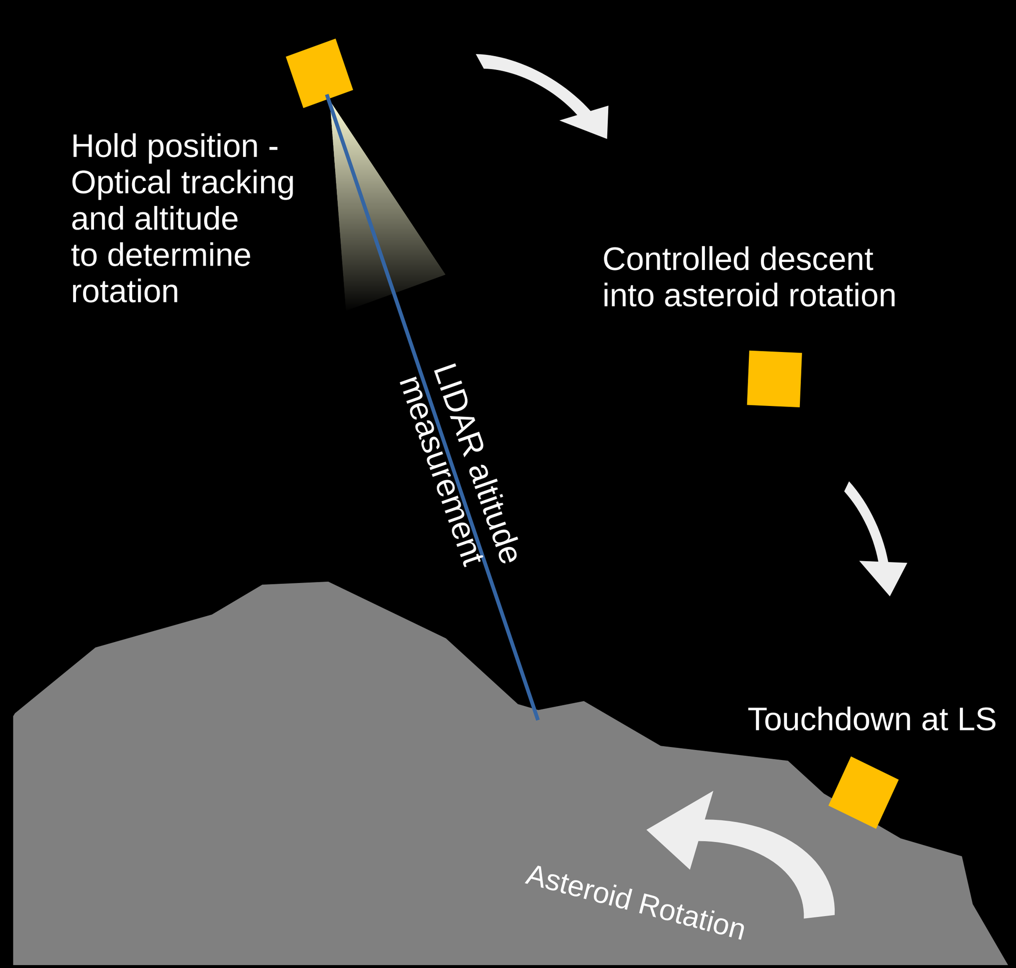

Fig 1. Overview of flight methodology

Implementation

A Zero-Effort-Miss/Zero-Effort-Velocity (ZEM/ZEV) based guidance system is used for thruster input control. This method calculates the distance by which the lander will miss the target if there are no changes to the current trajectory of the lander or of the landing site (LS) at the expected time of interception. An acceleration vector is calculated to correct for this error by the Guidance, Navigation and Control system (GNC). The velocity at interception time is considered in the same manner, where our target velocity is the linear velocity of the surface at the LS.

To accurately predict where the LS will be at interception time, we perform feature matching on optical data captured by the landers onboard camera in order to estimate rotation. Figure 2 provides an overview of the modelled lander components.

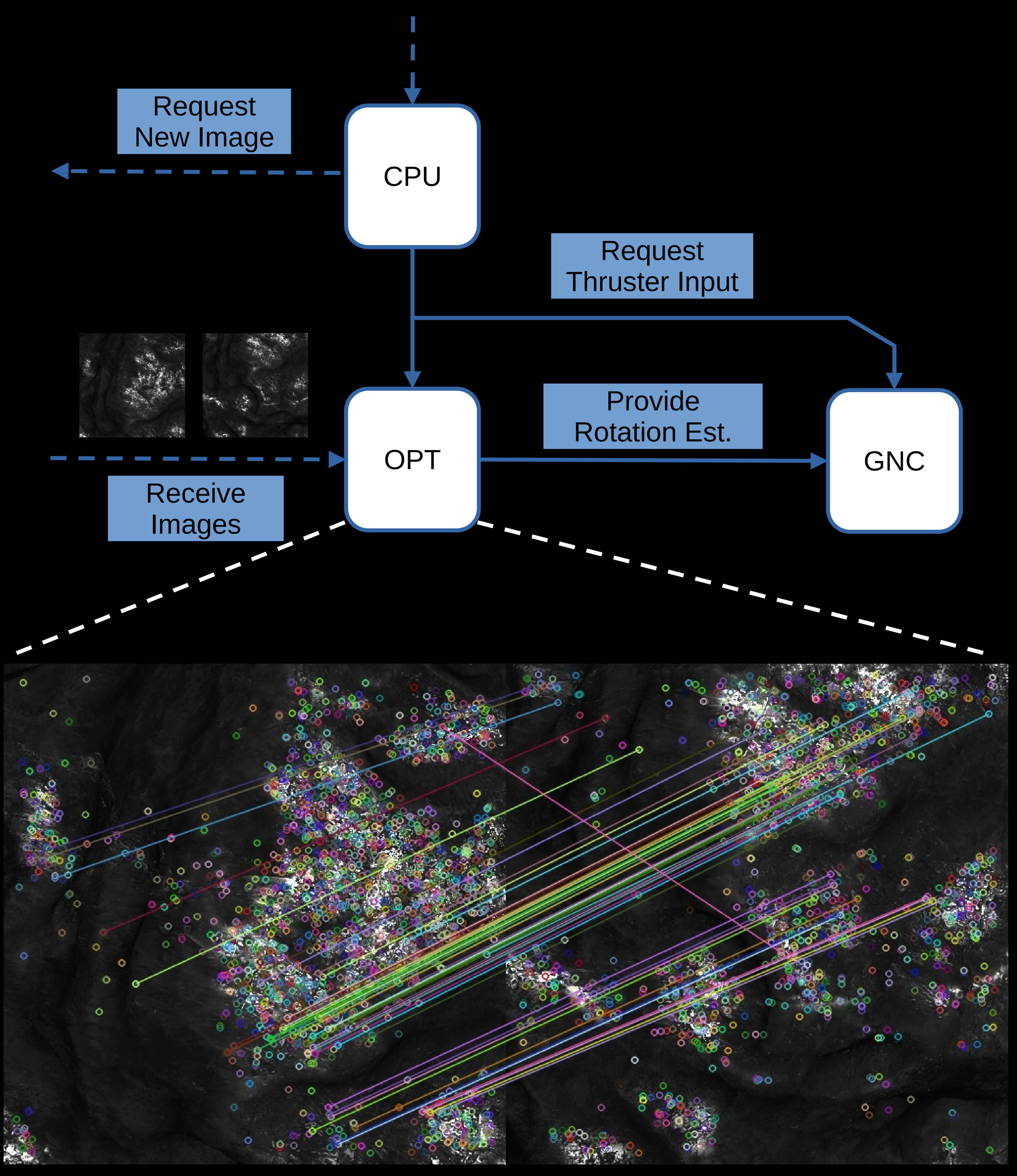

Fig 2. Lander logic and control diagram with SURF matching

The Optics (OPT) component processes incoming images and detects features. Once two sets of features are available they are matched between images, shown above. We then derive the homography matrix from this set which mathematically describes the relationship between the points on the images. With homography decomposition we extract the rotation and translation components from the matrix to generate an estimation of rotation. The rotation estimation is then passed to the GNC and used to predict when to begin the approach phase and used to estimate the LS position at any given time.

Results

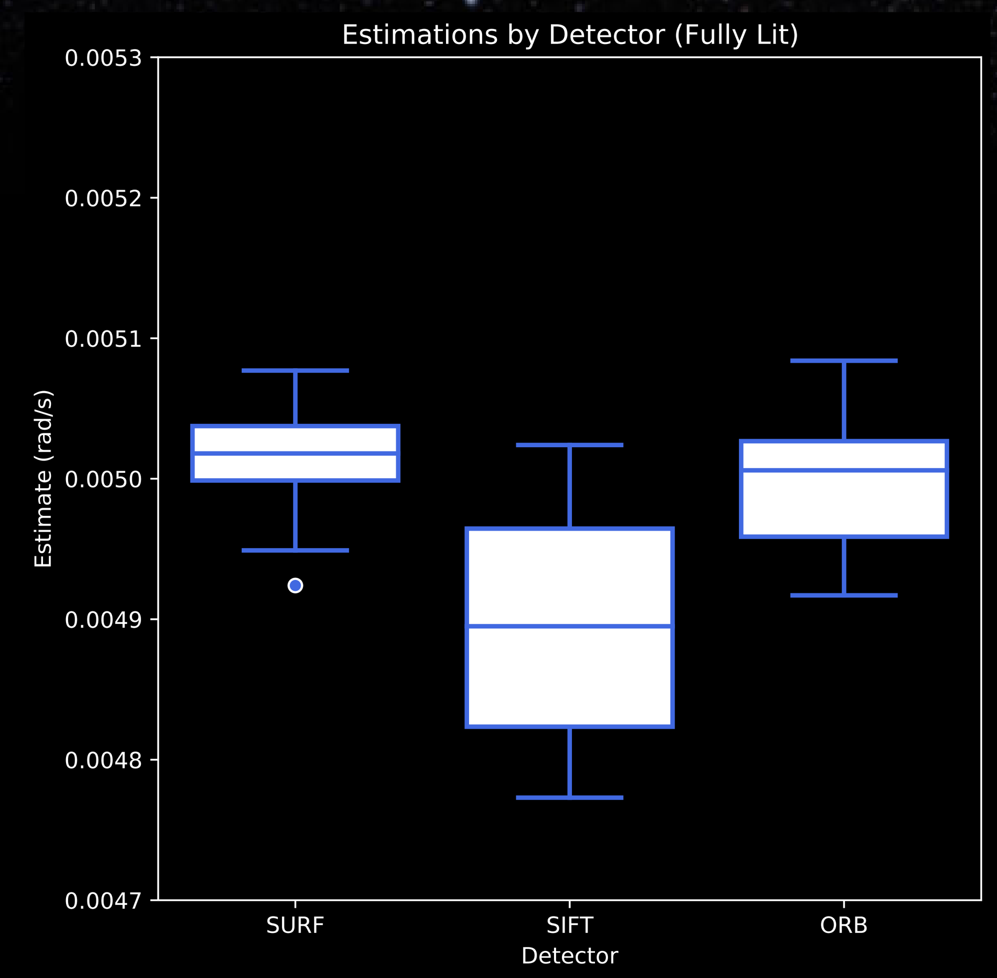

Fig 3. Feature detection comparison

Comparing detection algorithms in a fully lit environment with 0.005 rad/s rotation at 10 estimations per detector. SURF has the tightest spread with a standard deviation of 0.0026 degrees/s. SIFT has a much higher spread and consistently underestimates with a SD of 0.0046 degrees/s. This appears to be due to a more localized selection of features matched from across images compared to SURF and ORB. SIFT in fact performs more favourably with 45 degrees of illumination due to large shadows limiting the amount of features available for all detectors.

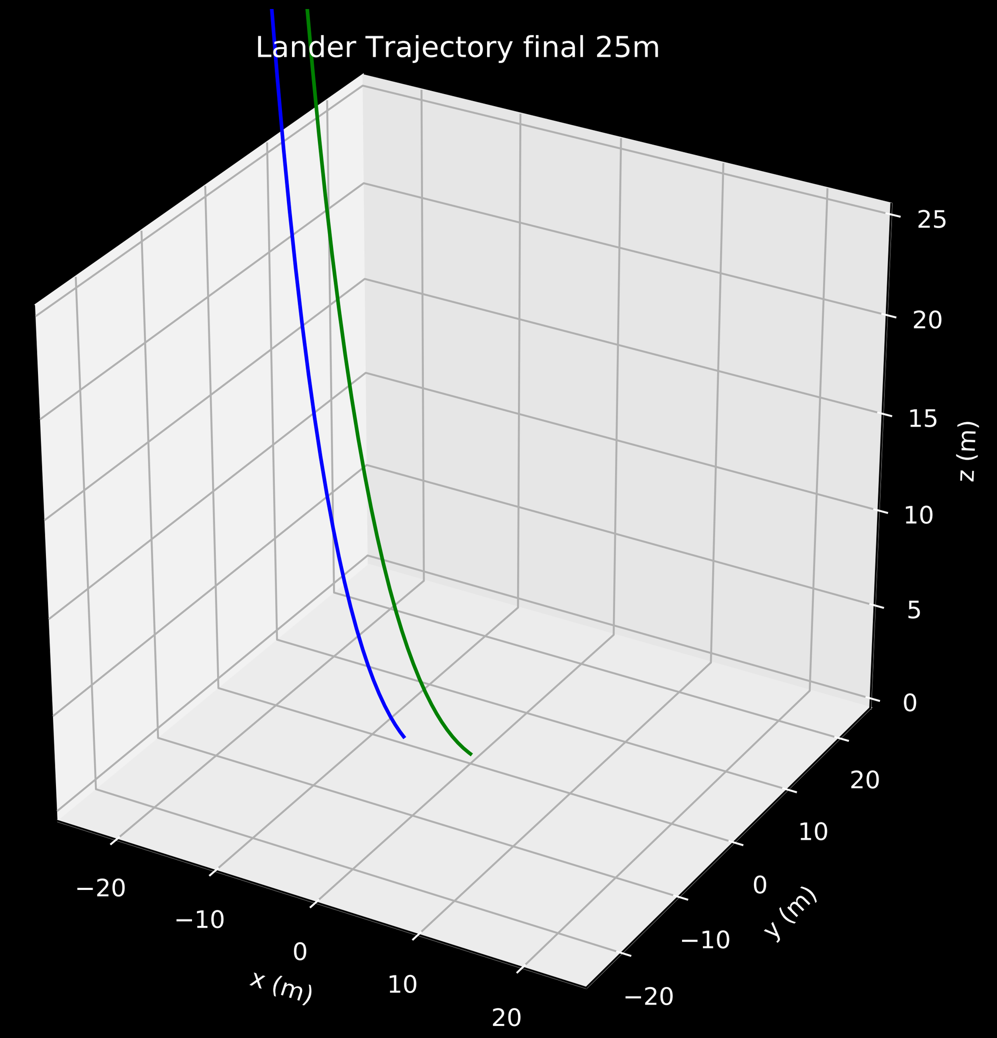

Fig 4. Final 25m of flight trajectory

Ideal flight path shown in green, actual flight shown in blue, demonstrating a typical underestimate of rotation. Over 1000 seconds of flight time, the deviation is minimal yet constant at ~6m.

ROBERT GORDON UNIVERSITY ABERDEEN
SCHOOL OF COMPUTING

Computer Science