Selecting Martian Landing Sites

Authors: Yago Botella Barbeito & Jaime Villarrubia Fernández

Subject: Python Programming

Date: 27/05/2025

1. Abstract

Landing a spacecraft on Mars presents numerous technical challenges due to the planet's harsh environment and thin atmosphere. In this project, we simulate the role of NASA engineers who must identify suitable candidate zones for Martian landings. By using satellite elevation data from the MOLA mission (Mars Orbiter Laser Altimeter), we apply statistical analysis in Python to locate regions near the equator that are flat, low in elevation, and therefore optimal for landing. Our solution utilizes NumPy, OpenCV, and tkinter to process image data, compute surface metrics, and visually represent the selected safe zones. The result is a functional and insightful tool that identifies the 20 best areas for potential landing missions.

2. Introduction

2.1 Aim

The goal of this project is to use Python programming to develop a tool that can analyze Martian elevation data and identify the safest potential landing zones for a spacecraft. The primary concern is operational safety, prioritizing low elevation and flat terrain, rather than geological interest.

2.2 Scope

The analysis focuses exclusively on the region of Mars located between 30° North and 30° South latitude, as these equatorial zones provide the warmest temperatures and the greatest solar exposure, ideal for powering landers. Elevation and terrain smoothness are the only criteria considered for selecting landing regions.

2.3 Methodology

We use a grayscale elevation map of Mars derived from the MOLA dataset. Our program slices this map into overlapping rectangular sections of 670 km by 335 km and computes statistical properties such as standard deviation and peak-to-valley difference for each rectangle. Regions with a mean elevation above a threshold are discarded. The program identifies the top 20 regions for each statistical measure, and the overlap between these two sets defines the final list of high-grade candidate zones. The selected zones are displayed visually on both the grayscale and color MOLA maps using a tkinter-based graphical interface.

3. Theoretical Background

Landing safely on Mars requires extensive consideration due to its low atmospheric density. The Martian atmosphere is only about 1% as dense as Earth's, which limits its ability to slow descending spacecraft. Thus, landing zones must be at the lowest possible elevation to maximize the amount of atmospheric drag available.

Moreover, since most landers rely on solar panels, sites near the equator are preferred due to more consistent sunlight and higher average temperatures. Terrain flatness is essential to avoid damage from rocks or uneven surfaces and to ensure solar panels and other mechanisms can operate properly.

To evaluate smoothness, two statistical metrics are used:

Standard Deviation (StD): Measures how spread out elevation values are. Low StD means the terrain is mostly flat.

Peak-to-Valley (PtV): Measures the difference between the highest and lowest elevation in the region. Low PtV indicates fewer sharp drops or rises.

These measures are calculated over each rectangle, and the best candidates are selected by ranking them according to their smoothness.

4. Practical Implementation

4.1 Data Preparation

We work with two main images:

A grayscale elevation image (`mola_1024x501.png`) used for calculations.

A color shaded relief image (`mola_color_1024x506.png`) used for final visualization.

Using OpenCV, we load both images into NumPy arrays. The program then converts the map dimensions from kilometers to pixels, based on Mars' circumference, and defines the dimensions of each search rectangle accordingly.

4.2 Region Analysis

We implement a class `Search`, which handles image slicing and statistical computation. For each rectangle:

* The mean elevation is checked against a threshold (e.g., 55 grayscale value).

* If valid, StD and PtV are calculated and stored.

Overlapping rectangles are used to avoid aliasing problems, ensuring we do not miss potential zones due to fixed step sizes.

4.3 Filtering and Ranking

After analyzing all rectangles:

* We sort the rectangles by their StD and PtV values separately.

* The top 20 rectangles in each category are selected.

* We determine the final list by intersecting the two sets (regions appearing in both lists).

4.4 Visualization

Using `tkinter`, we create a graphical interface that displays the color MOLA map with the final high-graded rectangles drawn. Each rectangle is annotated with its number and statistical values (mean, StD, PtV). This final screen allows easy interpretation and validation of results.

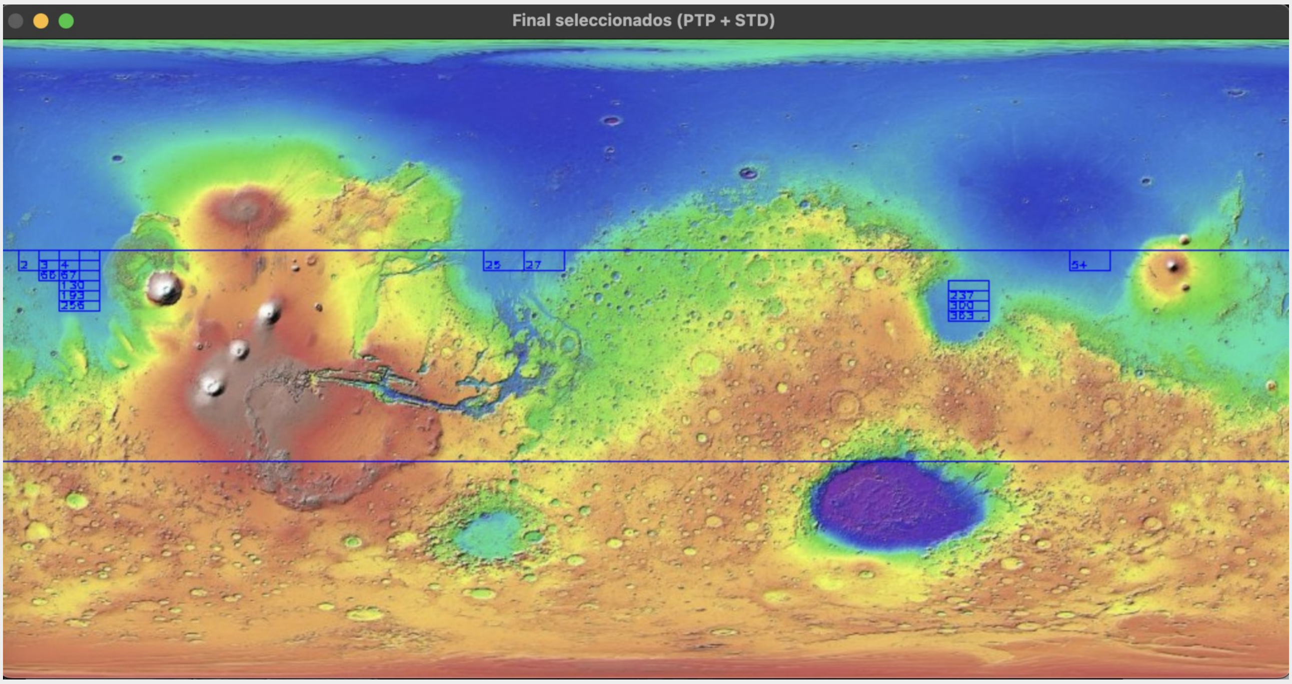
5. Summary

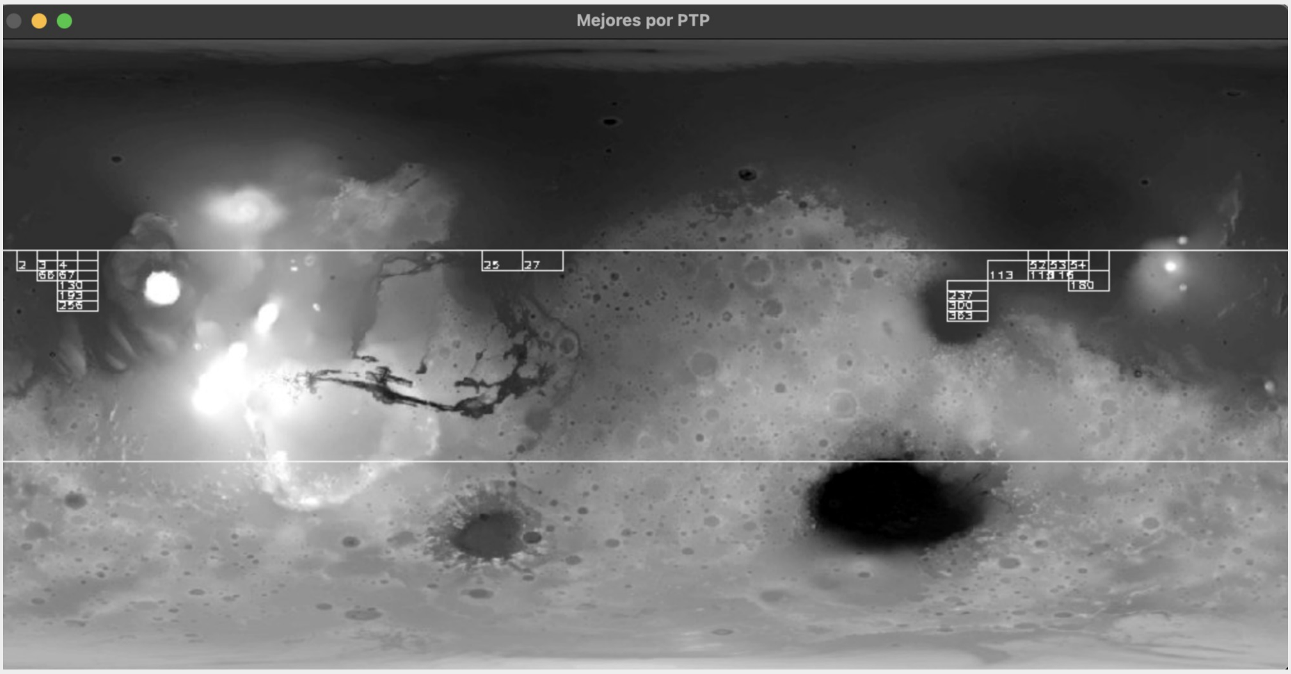
The project successfully replicates a realistic and educational scenario where Python programming is used to solve a scientific problem. By applying image processing and statistical analysis, we generated a set of candidate Martian landing sites that meet critical safety requirements. This project demonstrates the interdisciplinary power of Python to merge space science, mathematics, and programming.

Additionally, this work helped us strengthen our skills in:

* Object-oriented programming (OOP)
* Image analysis using OpenCV
* Numerical computing with NumPy
* GUI development with tkinter

It also provided insight into how NASA engineers might begin the complex process of selecting landing zones for future missions.






```
~/Documents/Yago/UNI/2º Cuatri/Python Programming/AdditionalFiles/r07
> python site_selector.py

=== Regiones seleccionadas ===
Rect#4 | Mean: 48.0 | Std: 0.00 | PTP: 0.0
Rect#67 | Mean: 48.0 | Std: 0.37 | PTP: 4.0
Rect#3 | Mean: 48.1 | Std: 0.57 | PTP: 4.0
Rect#25 | Mean: 48.2 | Std: 1.31 | PTP: 12.0
Rect#237 | Mean: 53.9 | Std: 1.50 | PTP: 10.0
Rect#256 | Mean: 52.9 | Std: 1.64 | PTP: 9.0
Rect#130 | Mean: 48.9 | Std: 1.65 | PTP: 7.0
Rect#66 | Mean: 48.9 | Std: 1.79 | PTP: 9.0
Rect#300 | Mean: 52.4 | Std: 1.87 | PTP: 10.0
Rect#193 | Mean: 50.8 | Std: 2.09 | PTP: 8.0
Rect#2 | Mean: 49.5 | Std: 2.13 | PTP: 9.0
Rect#363 | Mean: 51.1 | Std: 2.15 | PTP: 9.0
Rect#27 | Mean: 50.0 | Std: 2.23 | PTP: 13.0
Rect#54 | Mean: 42.4 | Std: 2.42 | PTP: 11.0
2025-05-20 11:58:57.064 Python[8127:830919] +[IMKClient subclass]: chose IMKClient_Modern
2025-05-20 11:58:57.064 Python[8127:830919] +[IMKInputSession subclass]: chose IMKInputSession_Modern
```

6. Bibliography

* Vaughan, Lee. *Real-World Python*, Chapter 7

* NASA Jet Propulsion Laboratory (JPL): MOLA Mission

* https://mars.nasa.gov/insight/

* https://astrogeology.usgs.gov/

* OpenCV Documentation: https://docs.opencv.org/

* Python Imaging Library (PIL): https://pillow.readthedocs.io/

* NumPy Documentation: https://numpy.org/

* Tkinter Documentation: https://docs.python.org/3/library/tkinter.html