Horizon-Based Optical Navigation (OpNav)

by Aaryan Sonawane

June 23, 2025

Report Goals

- Estimate the intrinsic calibration parameters for the LORRI camera (neglect radial/tangential distortion).
- Determine the position of New Horizons relative to Pluto in the camera frame using horizon-based OpNav.

1 Camera Calibration

The pinhole camera model projects a 3D point $[X, Y, Z]^T$ to image-plane coordinates $[x, y, 1]^T$ via:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \propto \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \tag{1}$$

To convert to pixel coordinates (u, v) we use the intrinsic matrix K:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} d_x & 0 & u_p \\ 0 & d_y & v_p \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \tag{2}$$

where

$$d_x = \frac{f}{\mu}, \quad d_y = \frac{f}{\mu}, \quad (u_p, v_p) = (\tfrac{W}{2}, \tfrac{H}{2})$$

and f is the focal length, μ the pixel pitch, and (W, H) the image dimensions. For LORRI: $f = 2630\,\text{mm}$, $\mu = 0.013\,\text{mm}$ gives

$$f_{\text{pix}} = \frac{2630}{0.013} \approx 2.023 \times 10^5 \text{ px}.$$

2 Horizon-Based OpNav

Pluto is modeled as a sphere of radius $R_P = 1188.3\,\text{km}$. An image of its limb has apparent pixel radius r_{pix}. From similar triangles:

$$\frac{r_{\text{pix}}}{f_{\text{pix}}} = \frac{R_P}{Z} \quad \Longrightarrow \quad Z = \frac{f_{\text{pix}} R_P}{r_{\text{pix}}}. \tag{3}$$

If the fitted limb center is (u_0, v_0) and principal point is (u_p, v_p), then:

$$x = (u_0 - u_p)\frac{Z}{f_{\text{pix}}}, \quad y = (v_0 - v_p)\frac{Z}{f_{\text{pix}}}.$$

The spacecraft camera-frame position is

$$\mathbf{r}_{\text{sc}} = [x,\, y,\, Z]^T.$$

3 Results

Figure 1 overlays the detected Pluto limb on the LORRI image. By fitting a circle with the Circle Hough Transform (CHT) I measured the limb's pixel radius r_{pix} and center coordinates (u_0, v_0). Substituting these into the similar-triangles equation (3) yields:

$$x = 45.70\,\text{km}, \quad y = -291.80\,\text{km}, \quad Z = 1{,}422{,}498.41\,\text{km}.$$

A Python Implementation

```
from google.colab import files
uploaded = files.upload()
import cv2
import numpy as np
import matplotlib.pyplot as plt

# Known parameters
R_pluto = 1188.3       # Pluto radius [km]
f_mm = 2630            # LORRI focal length [mm]
pixel_pitch_mm = 0.013 # LORRI pixel size [mm]

# Compute focal length in pixel units
f_pix = f_mm / pixel_pitch_mm

# Load and preprocess image
filename = next(iter(uploaded))
img = cv2.imread(filename)
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

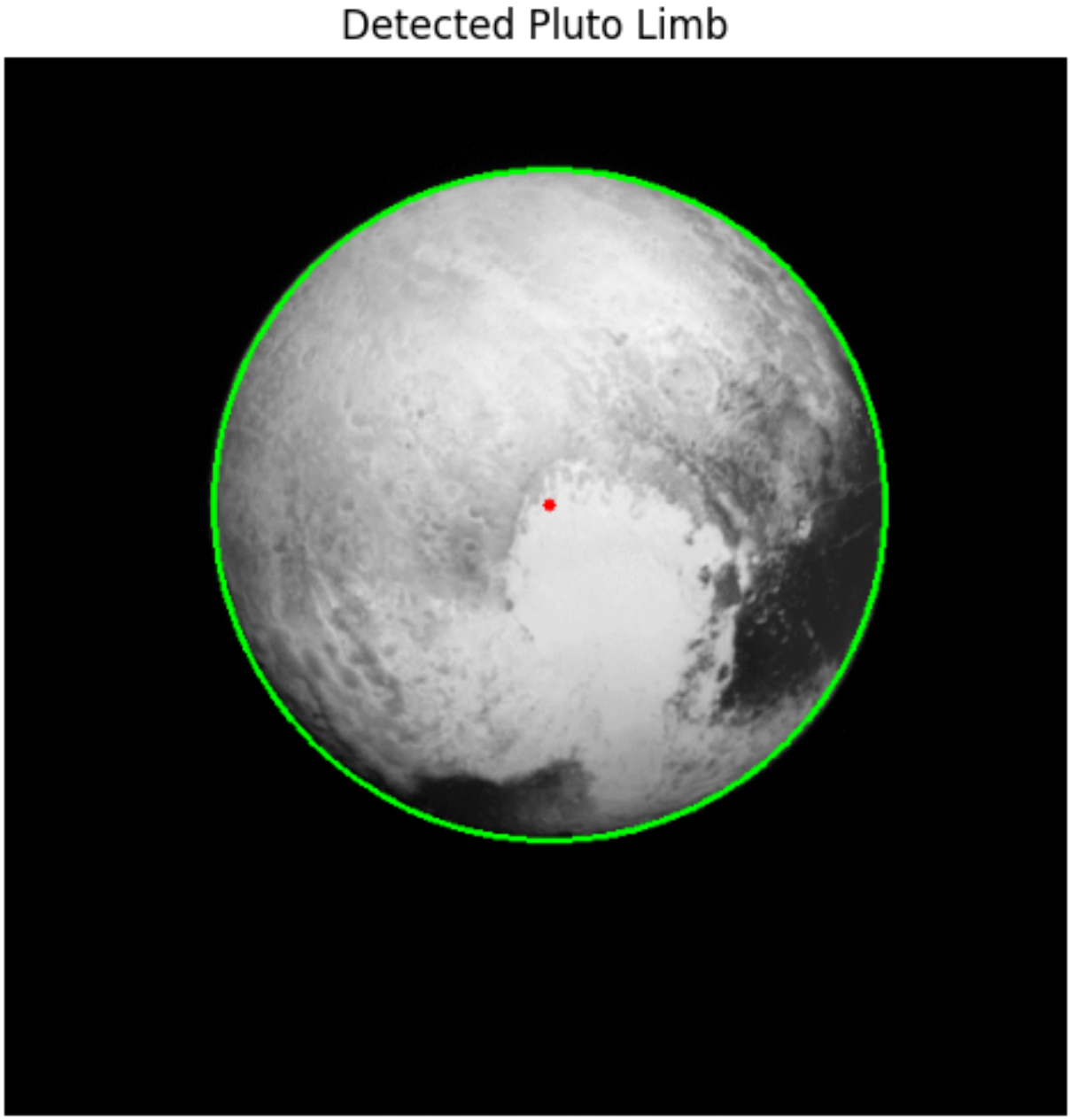

Figure 1: Detected Pluto limb via CHT. Green circle is the fitted limb and the red dot is the center.

```
gray = cv2.medianBlur(gray, 5)

# Image center
h, w = gray.shape
up, vp = w / 2.0, h / 2.0

# Detect limb via CHT
```

```
circles = cv2.HoughCircles(
    gray,
    cv2.HOUGH_GRADIENT,
    dp=1.2,
    minDist=200,
    param1=100,
    param2=30,
    minRadius=100,
    maxRadius=600
)

if circles is not None:
    circles = np.round(circles[0, :]).astype(int)
    u0, v0, r_pix = circles[0]
    # Visualize detection
    disp = img.copy()
    cv2.circle(disp, (u0, v0), r_pix, (0, 255, 0), 2)
    cv2.circle(disp, (u0, v0), 3, (0, 0, 255), -1)
    plt.figure(figsize=(6,6))
    plt.imshow(cv2.cvtColor(disp, cv2.COLOR_BGR2RGB))
    plt.title('Detected Pluto Limb')
    plt.axis('off')
    plt.show()

    # Compute distance [km]
    Z_km = f_pix * R_pluto / r_pix

    # Compute 3D position [km]
    x = (u0 - up) * Z_km / f_pix
    y = (v0 - vp) * Z_km / f_pix
    r_sc = np.array([dx, dy, Z_km])
```