Feature Matching from Orbiting Vehicles

Nicholas Neel, Caleb Adams, David L. Cotten
Small Satellite Research Laboratory, University of Georgia,

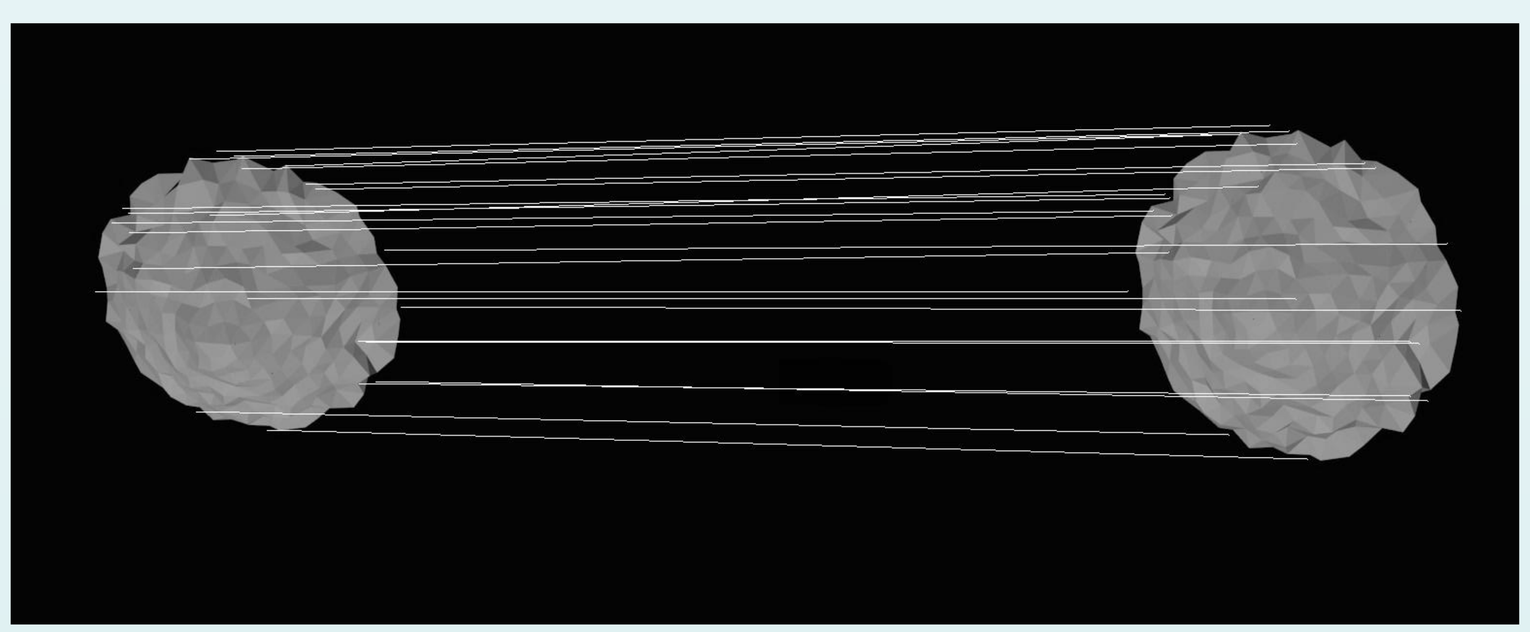
Overview

Feature matching offers an interesting problem for running Structure from Motion analysis from orbit. The root of this deals with all of the constraints introduced onto both the satellite and a feature on the surface of the object the satellite is rotating about. This is vital in applications such as the University of Georgia Multiview On-board Computational Imager (MOCI). In order to create a sparse point cloud at a location of interest, we need to find matched keypoints to use with the multi-view geometry. To explore this fully we need to take into account the feature detection algorithm's outputs, and pair that with the flight path of orbiting spacecraft.

Scale Invariant Feature Transform (SIFT)

The Scale Invariant Feature Transform identifies keypoints in an image by finding local extrema in scale space using difference of gaussians. The output of this algorithm is typically the scale, orientation, and a 128 dimensional vector of each identified keypoint. In order to match keypoints, we simply compute the euclidean distance between two keypoints and create a tied point if their euclidean distance is small enough. In order to avoid the brute force method for matching, we will develop filters to place on pixels in order to estimate which area in the next image has a highest probability to occur.

Simulating the Orbital Imaging Process



An image set using the University Small Satellite Research Laboratory's blender suite for simulating satellite imagery. Using these images we are able to see how a keypoint shifts while point tracking an object. Assuming an equatorial orbit, that is to say an orbit in which the axis of rotation for the object is in the plane's normal direction, a target centered on the equator, a normal facing camera, and an orbiting body in a perfectly circular orbit, we can use the following equations. The first is the position of the satellite, p(t), at time t, and the position of the feature, at time t.

$$p(t)=(r\cos(\omega t), r\sin(\omega t), 0)$$
$$f_{xyz}(t)=\left(\sqrt{x^2+y^2}\,\cos(\omega_e t), \sqrt{x^2+y^2}\,\sin(\omega_e t), 0\right)$$

Image plane feature location

In order to map the feature to the image plane, we will compute the Frenet frame of the curve. This gives us a relative coordinate system for the satellite in order to find where the image is. Since we are assuming equatorial, circular orbit we obtain,

$$\hat{T}(t)=(-\sin(\omega t), \cos(\omega t), 0)$$
$$\hat{N}(t)=(-\cos(\omega t), -\sin(\omega t), 0)$$
$$\hat{B}(t)=(0,0,1)$$

Now we just need to calculate the angle from the Normal and Binormal vectors from the feature relative to the satellite's position. Next, we need to perform a change of basis calculation in order to find the relative angle to the feature from the satellite so we can match the feature onto the image plane. To do this we utilize the following equation.

$$f'_{xyz}(t)=f_{xyz}(t)-p(t)$$

Now we can determine where the feature is on the focal plane by finding the angles in the T,N plane and in the N,B plane. This will give us the angles of the keypoint from the camera. Namely,

$$x_{kp}(t)=f\,\frac{\langle f'_{xyz}(t), \hat{T}(t)\rangle}{\langle f'_{xyz}(t), \hat{N}(t)\rangle}$$
$$y_{kp}(t)=f\,\frac{\langle f'_{xyz}(t), \hat{B}(t)\rangle}{\langle f'_{xyz}(t), \hat{N}(t)\rangle}$$

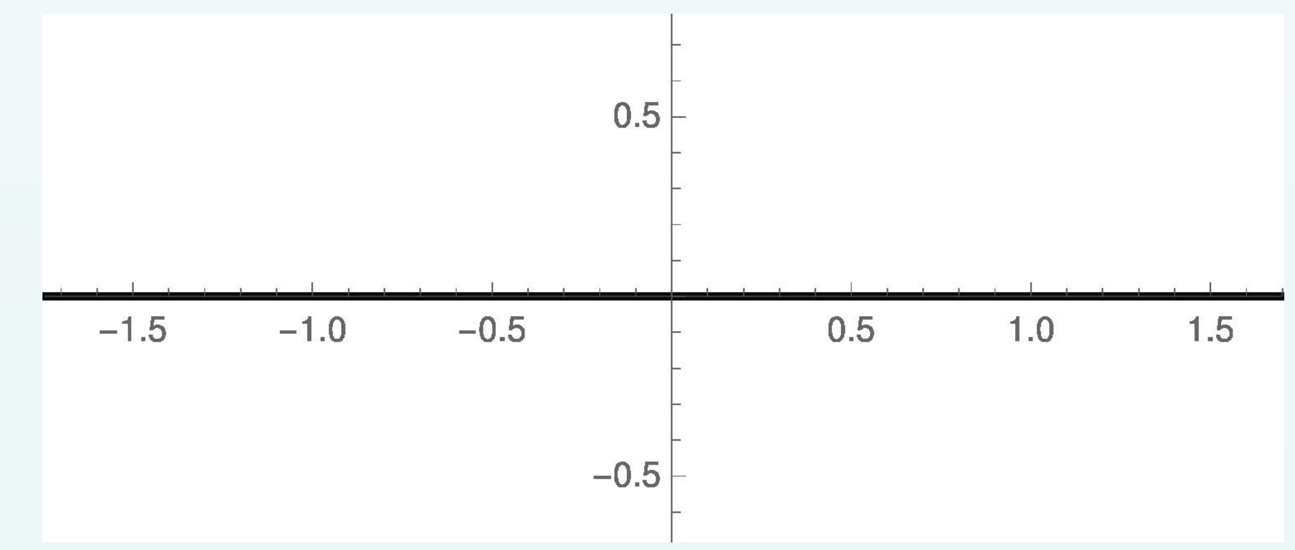
This gives our equation for the path of the feature as:

$$F(t)=(x_{kp}, y_{kp})$$

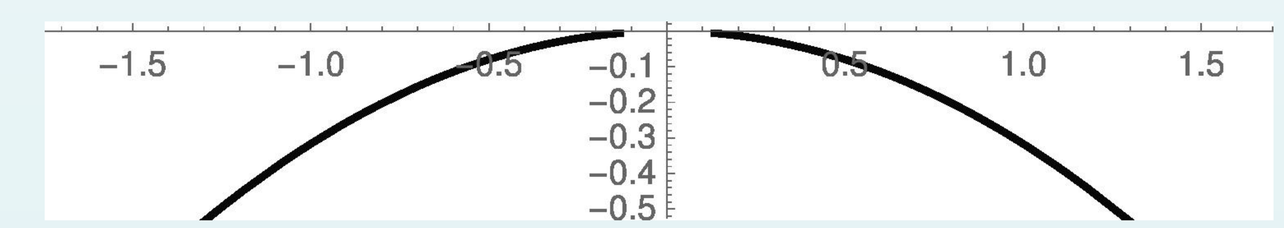
Using this method we can observe what kinds of paths to expect from a pixel. Future work of this is to generalize to any orbit, planetary body, and to possibly expand to point tracking equations. The advantage of this approach is that it offers a direct way to compute a feature path. The way to take advantage of this is by using a points current location to estimate where it will be in the next image.

Results

Below are several keypoint locations being tracked throughout the orbit.

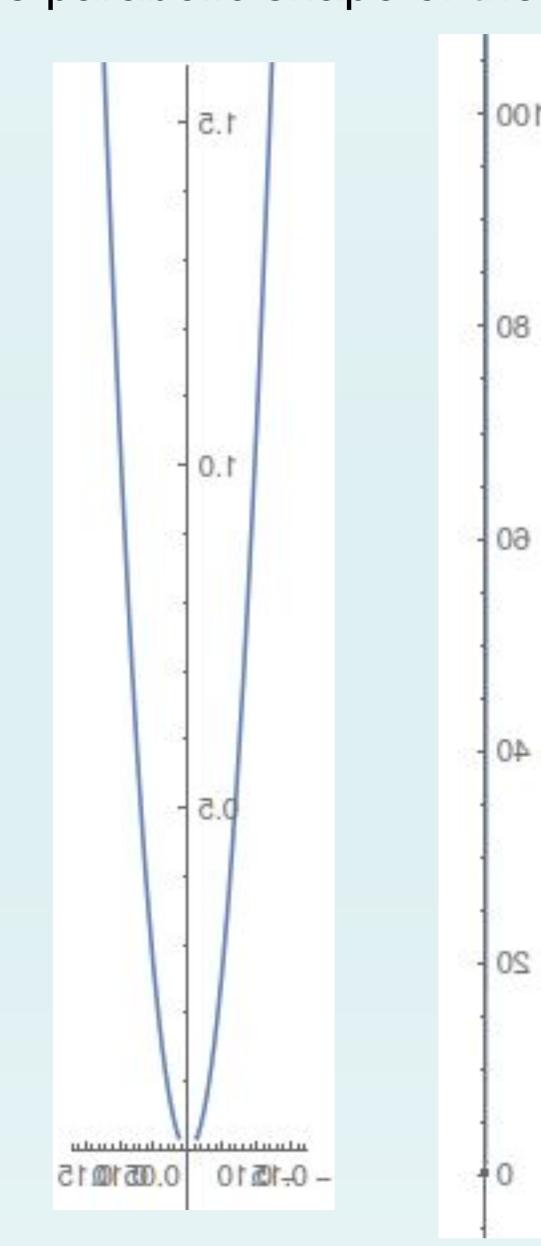


This first figure is a point at the equator. The result of this as expected is just a line in the camera. This type of feature is consistent with epipolar geometry.



This first figure is a point north of the equator. Note that the image coordinates need to be converted to real pixel coordinates, but notice the parabolic shape of the feature.

The Figures to the right are increasing z values. As the distance from the equator increases, the sharper the parabola is.



For more information about CubeSats, the Small Satellite Research Laboratory (SSRL), our missions, visit www.smallsat.uga.edu.