Aligning the Stars - Aurora Localization via Star-Trails

Laura Culp, Fangda Li, Andre Recnik

April 29, 2013

Abstract

This is a brief description of the methods and the math behind the methods used and attempted when trying to solve the problem outlined for the NASA SpaceApps Hackathon in Toronto, Ontario.

1 Introduction

There are many time-lapse picture sequences taken from the International Space Station (ISS) that include auroras. Here, we attempt to take these videos and extract, for each frame, the location of the aurora on the earth. The only information given is the location (Nadir point and altitude) of the ISS at the time that each picture was taken.

2 Determining Camera Orientation

One of the most interesting parts of this problem is the determination of the orientation of the camera. Given the NADIR point of the ISS and the altitude, we can calculate the circle of possible areas the picture was taken of, where the ISS is at the center. The problem is that we do not know where in the circle the picture was taken. If we can determine the angle of the camera relative to the orbit of the ISS, the star patch that we are looking at, or the longitude that is being viewed, we are able to work out the angles.

There were three methods attempted. The first method, star matching via least squares, was fully implemented and determined to be too inefficient. The second method, tracking the velocity of landmarks on the earth to determine longitude, was determined to not be robust while completing a rough trial. The last method, building star-trails and determining the velocity of the stars and thus the angle of the camera off the orbit of the ISS, worked. Each method is described briefly below.

2.1 Star Matching via Least Squares

2.2 Tracking Velocity of Earth

The matlab script ’$match_features.m$’ shows one of the methods that was used to track landmarks on the earth. SURFT, SIFT, FAST, and Harris features were all attempted. An exhaustive search of the parameter space for these functions was not completed, and so matching features on the earth still could work, but what was seen in the trials that were completed was a lack of correct matching of enough features.

2.3 Star-Trails

3 Computational Method Outline

A brief description of the method that was used to determine the orientation of the camera relative to the International Space Station, which stays constant for an entire video sequence, is as follows:

1. Segment images into ’earth’ and ’sky’ using k-means
2. Extract the brightest stars from each frame
3. Stack extracted stars to form star-trails

4. Calculate the length of each star-trail and convert to velocity data
5. Determine the angular offset of the camera from the ISS orbit

Once the orientation of the camera relative to the International Space Station has been determined, the approximate location of the aurora in each image is placed on a map using the following method:

1. Approximate where the aurora is in the image via segmentation
2. Calculate the area of the earth visible in the image
3. Project the approximate location of the aurora onto the earth

The details of these methods will be explained further below.

3.1 Image Segmentation

In order to avoid incorrect identification of stars in the images, the area in the image belonging to the earth must be separated from the area in the image belonging to the sky. The location of the aurora itself must also be approximated in the image in order to be able to map it to a location on the earth. This can be done using a computer vision technique called segmentation.

There are two methods of image segmentation that were attempted: k-means and level sets. Each method worked, but the k-means algorithm worked more efficiently and was able to provide a segmentation of both the aurora and the earth at the same time. These methods are briefly described.

3.1.1 k-means

K-means segmentation is a clustering technique which separates the image into a given number, k, of clusters, each centered around a Gaussian distribution. There is an efficient implementation of this algorithm included in a library in matlab, 'kmeans'. This implementation was used.

In order to more-accurately identify the clusters, the entire image was scaled down and blurred. This avoids anomelies being placed in the incorrect clusters.

3.1.2 Level Sets

The level set method tracks an evolving contour line. The approximate level sets are tracked, and when the slope of the contour line is greater than a given parameter, the current contour line is used to determine a cut in the image. This only separates the image into two different sections, and can take slightly longer than the k-means algorithm. In addition, there are multiple parameters that need to be played with and correctly chosen in order to get the desired results.

An implementation of this method given by Jeff Orchard in a Medical Image Processing course at the University of Waterloo was used.

3.2 Star-Trail Creation

3.3 Velocity Extraction

3.4 Camera Orientation Calculations

3.5 Projections and Display of Map

At this point we have the orientation of the camera, and the approximated aurora and earth segmentation data. We now need to calculate the area of the earth visible from the camera, and from that, approximate the aurora location.

3.5.1 Earth Visible from Camera

We know the orbit of the ISS. Let the current position of the ISS be x and the position of the ISS in the next frame be x'. We also know the angle of the camera off the axis of the ISS orbit, let it be θ.

Then, we can approximate the direction of the camera to be

$$camera = \begin{pmatrix} \sin\theta & -\cos\theta \\ \cos\theta & \sin\theta \end{pmatrix} \frac{(x-x')}{norm((x-x'))}$$

We can also calculate the distance to the horizon from the ISS as follows

$$distancetohorizonfromISS = \sqrt{(RadiusofEarth + ISSAltitude)^2 - (RadiusofEarth)^2}$$

and the distance to the horizon point along the ground as

$$distancetohorizonalongearth = arcsin(\frac{(distancetohorizonfromISS)*\sin\frac{\pi}{2}}{RadiusofEarth+ISSAltitude}$$

Then, since the horizontal field of view of the camera on the ISS is 60 degrees, the leftmost point of the horizon seen from the ISS is calculated to be

$$vectortoleftmosthorizonpoint = \begin{pmatrix} \sin\frac{\pi}{6} & -\cos\frac{\pi}{6} \\ \cos\frac{\pi}{6} & \sin\frac{\pi}{6} \end{pmatrix} (camera)$$

$$leftmosthorizonpoint = \frac{distancetohorizonalongearth}{distance(x, vectortoleftmosthorizonpoint + x)} * vectortoleftmosthorizonpoint$$

Where distance calculates the distance between two (lat, lon) coordinates in kilometers.

A similar calculation can be done for the rightmost point on the horizon that is seen on the camera.

This includes many approximations, and assumes that the camera is always level with the horizon, but gives a rough guess of the area on the earth that the camera could see.

3.5.2 Approximate Aurora Location

We now can project each pixel that is guessed to be the aurora back onto the wedge of the earth that is visible to the camera.

First, we calculate which approximate row in the image is the horizon. This does not account for any curvature along the horizon, and is an approximation. It is done by calculating:

$$horizonrow = \left\lfloor \frac{numofpixelssegmentedtobethesky}{totalnumofpixels} * numberofrows \right\rfloor$$

Then, given some point in the image, we can project it onto the map by finding the vector pointing the correct direction by looking at the offset of the pixel from the horizontal center and rotating it by a fraction of a degree (the horizontal field of view is 60 degrees). We can then find the distance to the point on the earth by taking into consideration the offset of the pixel from the horizon and the vertical field of view (45 degrees). We combine this in a method similar to what is shown above.

The method '$ll_of_xy_in_image$' implements this.

4 Conclusion

Though there are many approximations in this method and many areas that could be made more robust, this method is able to calculate the approximate location of an aurora over the earth in an efficient manner using only a series of time-lapse photos from the ISS and the ISS location when each photo is taken.

5 Acknowledgements

Our team wishes to thank Jeff Orchard for the implementation of Level Sets Segmentation and the wonderful Medical Image Processing course notes, the ISS Crew for the beautiful pictures, and the Toronto SpaceApps Team for organizing the Toronto Nasa SpaceApps Challenge.