

# Aligning the Stars - Aurora Localization via Star Trails

Laura Culp, Fangda Li, Andre Recnik

This is a brief description of the analytical and numerical methods used to solve the "Aligning the Stars" challenge outlined for the NASA SpaceApps Hackathon in Toronto, Ontario.

## I. INTRODUCTION

Auroras are visible in many time-lapse picture sequences taken from the International Space Station (ISS). Our goal is to use this data to determine the location of the aurora on the earth. The only additional information available is the location (Nadir point and altitude) of the ISS for each frame.

## II. DETERMINING CAMERA ORIENTATION

Given only the Nadir point of the ISS and its altitude, one can determine circular horizon centred on the ISS in which the visible aurora must be constrained. However, the camera's field of view covers only  $\sim \pi/3$ , so the problem becomes the determination of where this field of view wedge is centred.

An exhaustive astrometric search was first implemented and found to be computationally intractable since the search grid spacing in RA, DEC, and frame rotation angle must be much smaller than the inter-stellar spacing on the frame  $\ll 1$  degree. We employ a much more efficient method via star trail mapping to determine the velocity of the stars and thus the angle of the camera off the orbit of the ISS.

## III. COMPUTATIONAL METHOD OUTLINE

Our method was 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.

### A. 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.

### 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 anomalies being placed in the incorrect clusters.

### 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.

## B. Star-Trail Creation

The star trails are created by first extracting the stars from the segmented images by comparing the median of a 3x3 grid and the median of a 9x9 grid around each pixel. This extracts just the brightest stars.

Then the extracted stars from each image are added together to form the star trails. To make the velocity extraction process more robust, the images are not given an equal weights. The first 5 images are given a small weight, the middle images are given a larger weight, and the last 5 images are given a very large weight. This allows the velocity extraction process to detect what direction the trail is moving in, and if the trail is complete (contains both a start and end point).

## C. Velocity Extraction

To extract the velocity from the star trails we simply need to measure the star trail lengths, and divide by the time interval.

To do this we look at every pixel in the image, and when we find a part of a star trail we use a flood fill algorithm to expand the area to the full trail. Given this full trail, we can compute the minimum and maximum points, and measure the distance between them to get the approximate arc-length.

Since we changed the weight of the star values for the start and end of the trail, it is possible to check what the direction of the star trail is in. This also solves the problem of incomplete star trails, were the star exits or enters the frame, since such a trail will lack either a start or end point, or both.

## D. Camera Orientation Calculations

Let  $p$  be the pixel size and let  $f$  be the focal length of the lens. We define a virtual celestial sphere whose rotation axis is define by the ISS revolution axis. Given the size of the image frame  $2x0 \times 2y0$  pixels and the virtual declination of the centre of the frame and the frame rotation angle  $(d0, \theta)$ , we then have the tangent plane projection relations for the virtual right ascension  $a$  and declination  $d$  of a pixel  $(x, y)$  in the frame:

$$d = \text{Arcsin} \left[ \frac{\text{Sin}(d0) + \frac{p \cos(d0)(-x+x0+(y-y0)\cot(\theta))\text{Sin}(\theta)}{f}}{\sqrt{1 + \frac{p^2(-x+x0+(y-y0)\cot(\theta))^2 \text{Sin}(\theta)^2}{f^2} + \frac{p^2((x-x0)\cos(\theta)+(y-y0)\text{Sin}(\theta))^2}{f^2}}} \right], \quad (1)$$

$$a = \text{ArcTan} \left[ \frac{p((x-x0)\cos(\theta) + (y-y0)\text{Sin}(\theta))}{f \left( \cos(d0) - \frac{p(-x+x0+(y-y0)\cot(\theta))\text{Sin}(d0)\text{Sin}(\theta)}{f} \right)} \right]. \quad (2)$$

where we define the centre of the image to have right ascension  $a = 0$ .  $d0$  is the angular difference between the direction in which centre of the frame is pointed, and the direction of motion of the ISS. We analytically extend the above functions to extend their region of applicability over the polar regions of the virtual sphere where they are otherwise singular. Defining

$$K(d0, \theta, x, y) \equiv (f \cos(d0) + p \sin(d0)((-y + y0) \cos(\theta) + (x - x0) \sin(\theta)))^2 \left( 1 + \frac{p^2((x - x0) \cos(\theta) + (y - y0) \sin(\theta))^2}{(f \cos(d0) + p \sin(d0)((-y + y0) \cos(\theta) + (x - x0) \sin(\theta)))^2} \right) \quad (3)$$

$$J(d0, \theta, x, y) \equiv f^3 \left( \frac{f^2 + p^2(x^2 - 2xx0 + x0^2 + (y - y0)^2)}{f^2} \right)^{3/2} \left( 1 - \frac{(f \sin(d0) + p \cos(d0)((y - y0) \cos(\theta) + (-x + x0) \sin(\theta)))^2}{f^2 + p^2(x^2 - 2xx0 + x0^2 + (y - y0)^2)} \right)^{1/2}, \quad (4)$$

we differentiate the tangent plane projections with respect to  $x$  and  $y$ , obtaining

$$\frac{\partial a}{\partial x}(d0, \theta, x, y) = \frac{p}{K(d0, \theta, x, y)}(-f \cos(d0) \cos(\theta) + p(y - y0) \sin(d0)) \quad (5)$$

$$\frac{\partial a}{\partial y}(d0, \theta, x, y) = -\frac{p}{K(d0, \theta, x, y)}(p(x - x0) \sin(d0) + f \cos(d0) \sin(\theta)) \quad (6)$$

$$\frac{\partial d}{\partial x}(d0, \theta, x, y) = -\frac{p}{J(d0, \theta, x, y)}(fp(x - x0) \sin(d0) + \cos(d0)(p^2(x - x0)(y - y0) \cos(\theta) + (f^2 + p^2(y - y0)^2) \sin(\theta))) \quad (7)$$

$$\frac{\partial d}{\partial y}(d0, \theta, x, y) = \frac{p}{J(d0, \theta, x, y)}(fp(-y + y0) \sin(d0) + \cos(d0)((f^2 + p^2(x - x0)^2) \cos(\theta) + p^2(x - x0)(y - y0) \sin(\theta))). \quad (8)$$

The revolution of the ISS around the Earth only changes the virtual right ascension of stars in the image and not their virtual declination. Thus, given a star trail length  $(\Delta x, \Delta y)$  over some time  $\Delta t$  centred on  $(x_c, y_c)$ , the residual for a single data point is:

$$R(d0, \theta, x_c, y_c) = \left( \Delta x \frac{\partial a}{\partial x}(d0, \theta, x_c, y_c) + \Delta y \frac{\partial a}{\partial y}(d0, \theta, x_c, y_c) - \Delta t \frac{2Pi}{P} \right)^2 + \left( \Delta x \frac{\partial d}{\partial x}(d0, \theta, x_c, y_c) + \Delta y \frac{\partial d}{\partial y}(d0, \theta, x_c, y_c) \right)^2 \quad (9)$$

where  $P$  is the period of the ISS revolution. We minimize the sum of  $R$  over all star trails that we extract in the image to optimize the parameters  $(d0, \theta)$ .

Note that this procedure does not compute the angle of the camera with respect to the horizon (i.e. how much of the ground vs. sky is shown). This angle can however be easily computed using the segmentation of the images, and comparing the amount of ground vs. sky in the image.

### E. 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.

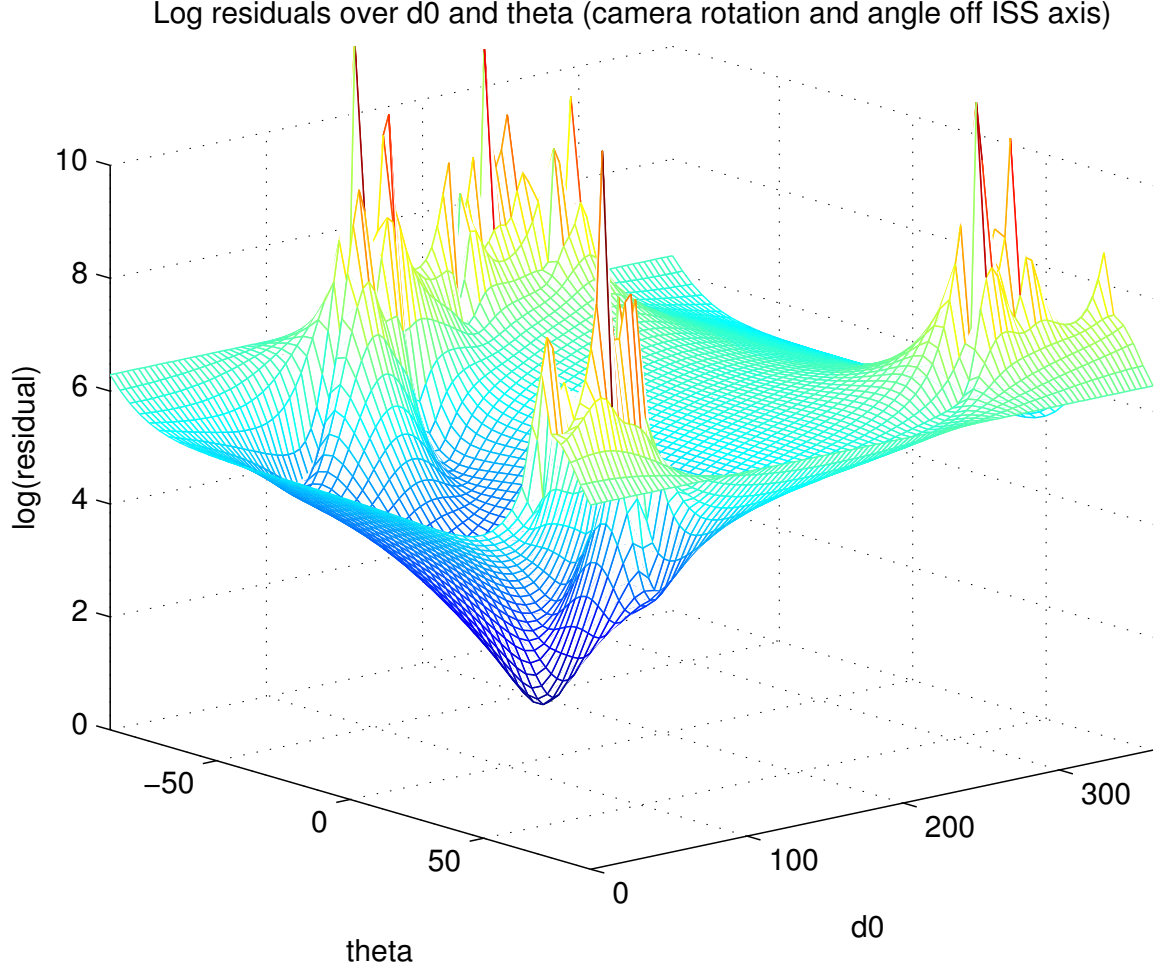


FIG. 1: Summed residual over all data points as a function of camera rotation angle  $\theta$  (deg) and virtual declination of the frame  $d0$  (deg).

### 1. Earth Visible from Camera

We know the orbit of the ISS. Let the current position of the ISS be  $\mathbf{x}$  and the position of the ISS in the next frame be  $\mathbf{x}'$ . We also know the angle of the camera off the axis of the ISS orbit  $d0$ .

Since the horizon distance is small compared to the radius of the Earth, we can approximate the direction  $\mathbf{m}$  of the camera as

$$\hat{\mathbf{m}} = \begin{pmatrix} \sin d0 & -\cos d0 \\ \cos d0 & \sin d0 \end{pmatrix} \frac{(\mathbf{x} - \mathbf{x}')}{|\mathbf{x} - \mathbf{x}'|}$$

The horizon distance is

$$h = \sqrt{(R_E + z_{\text{ISS}})^2 - (R_E)^2}$$

where  $z_{\text{ISS}}$  is the altitude of the ISS and the distance to the horizon point along the ground is

$$\tilde{h} = \arcsin\left(\frac{h \sin \frac{\pi}{2}}{R_E + z_{\text{ISS}}}\right)$$

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

$$\hat{\mathbf{v}}_L = \begin{pmatrix} \sin \frac{\pi}{6} & -\cos \frac{\pi}{6} \\ \cos \frac{\pi}{6} & \sin \frac{\pi}{6} \end{pmatrix} (\hat{\mathbf{m}}) \quad (10)$$

$$\mathbf{v}_L = \frac{\tilde{h}}{|\mathbf{x} + \hat{\mathbf{v}}_L|} \hat{\mathbf{v}}_L \quad (11)$$

where the distance between two latitude/longitude coordinates in kilometers. A similar calculation can be done for the rightmost point on the horizon that is seen on the camera. This assumes that the camera is always level with the horizon. The correct projection will be be shortly added to the algorithm.

## 2. Approximate Aurora Location

We now can project the pixel location of the aurora back onto the wedge of the earth that is visible to the camera. First, we calculate the approximate row  $M$  the horizon is in the image. This does not account for any curvature along the horizon, so is valid only in the linear regime:

$$M = \left\lfloor \frac{N_{\text{sky}}}{N_{\text{tot}}} N_{\text{rows}} \right\rfloor$$

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  $\sim 60$  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 ( $\sim 45$  degrees). We combine this in a method similar to what is shown above.

The method `'ll_of_xy_n_image'` implements this.

## IV. LIMITATIONS AND CONCLUSION

Although we have used the correct spherical and tangent plane projection from the celestial sphere to pixel coordinates for the optimization of the frame centre components  $(\theta, d_0)$ , we have not done so for the transformation back from pixel coordinates to coordinates on the globe itself. However, due to the small angles that this latter transformation uses, this accounts for a difference by a geometric factor  $\lesssim 20\%$  in most cases. We have demonstrated that 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.

## V. 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. Computational resources were provided by the Canadian Institute for Theoretical Astrophysics.