

EFFICIENT STREAK DETECTION AND PLATE-SOLVING TECHNIQUES

N. Houtz and C. Frueh

Purdue University, 610 Purdue Mall, West Lafayette, IN, United States, Email: {nhoutz, cfrueh}@purdue.edu

ABSTRACT

Optical ground-based observations of objects in Earth orbit produce images where light from stars and satellites may be spread over hundreds or thousands of pixels. This paper addresses the challenges of detecting and locating streaked objects with a new algorithm, and presents a new plate-solving technique that is robust for images where many stars may not be identified due to streaking. The image processing techniques are demonstrated on a series of 15 observations from a ground-based telescope. An average of 11.7 streaked stars were identified per image with an average of less than 1 arcsecond of error in star vectors.

Keywords: Space debris; image processing; optical observation; streak detection; plate solving; pattern matching.

1. INTRODUCTION

Optical observations of faint objects in Earth orbit typically use long exposure times to gather as much light as possible when tracking them. This causes stars to streak over many pixels in the image, which makes detecting and locating them difficult. Consistent detection and accurate location of stars is important to be able to properly identify them in a reference catalog and generate a precise attitude estimate, which is used to calculate the directions to other objects of interest in an image.

The standard method of detecting objects in black and white images is to apply a simple threshold to each pixel based on the noise level of the background [8]. The threshold is typically chosen to be several standard deviations above the mean background so that the probability of detecting random noise is small. There are several strategies to decrease the required threshold without increasing false positive detections. If the point-spread function (PSF) spreads over many pixels, detections may be filtered such that groups of neighboring pixels are detected together. Assuming noise in each pixel is independent of its neighbors, the likelihood that several neighboring pixels all exceed a threshold due to noise is small. Gaussian smoothing is also commonly used to reduce noise.

When streaks are expected in an image, the ability to detect objects can be improved by evaluating groups of pixels together. Zimmer et. al. created a method intended for Graphics Processing Units (GPUs) which uses a modified Radon transform over the whole range of possible angles (0° to 180°) and finds anomalies compared to what would be expected from noise [15]. Vananti et. al. developed a technique where the filter in the shape of a streak was approximated and convolved with an image [12]. By modifying the parameters of the filter, the best fits for each streak can be estimated. Lastly, Schneider et. al. derived a more rigorous mathematical model of a streak by convolving a Point Spread Function with a rectangular function [9], which was used in the algorithm given by Dawson et. al. [3]. This algorithm characterizes a streak by five parameters, and then searches the entire parameter space by computing a log-likelihood of every possible streak to occur in the image.

Once objects have been detected and located, the stars in an image are used to perform a plate-solving procedure. Plate-solving is similar to spacecraft attitude determination using a star tracker, except that a-priori knowledge is expected to be available as ground-based observatories can record their azimuth and elevation during an observation. Additionally, star trackers usually have very limited memory and computing requirements, so algorithms developed for them seek to minimize the size of the stored on-board catalog and simplify the matching strategy. Spratling and Mortari published a review of many star tracker matching algorithms in 2009 [11]. The most common strategy is matching angles between star vectors, or a similar property like the dot product, as is done in Mortari's Search-Less Algorithm [6]. Other methods match properties of planar [2] or spherical triangles [1] formed by star unit vectors. In 1997, Padgett and Kreutz-Delgado [7] published a unique pattern-matching algorithm, that projects star vectors onto a Cartesian grid. Cartesian grids from the image and the reference catalog must be aligned with each other to be matched, so the grid's x-axis is chosen to be the direction of the closest star to the central star. But because star streaks often cannot be precisely located in an image and are sometimes left out of the pattern matching process altogether, it is difficult to guarantee that the axis-defining star of the image pattern will be correctly located so that the patterns can be properly aligned for matching. A modified grid algorithm that uses a polar grid was proposed in 2007 by Lee et. al. [5]. With this method, the so-called "pivot

star” can be re-selected, increasing the probability that a matching pattern will be found. Other methods using information similar to polar coordinates for matching patterns have also been proposed that match the radial and cyclic information of each star separately [14, 13].

In this paper, we describe a detection method that uses a similar strategy to the line integration technique described in [15], but which allows pixels to be grouped into wider lines and uses cumulative sums to evaluate sections of any length within each line. Then, a polar grid pattern matching algorithm is presented that allows patterns to be matched even if the star that defines the pattern’s axis is different in the image pattern and catalog pattern. These techniques are demonstrated on a series of images taken with the POGS telescope.

2. METHODS

A 14-inch (35.6 cm) Corrected Dall-Kirkham telescope, referred to as the Purdue Optical Ground Survey (POGS) telescope is operated by the Space Information Dynamics (SID) research group at Purdue University to collect optical image data of objects in Earth orbit. Two of the processing techniques for interpreting data from these images are described and their effectiveness analyzed. Two types of observations are typical: sidereally tracked images with stationary stars, and object-tracked images where stars streak across the image. First, a novel strategy of efficiently identifying faint streaking objects in an image is described along with the method of distinguishing stars from other objects of interest. Second, a new plate-solving algorithm used to accurately determine the pointing direction (and therefore the directions to objects in images in the inertial reference frame) using large numbers of identified stars is detailed.

2.1. Image Processing

To quickly detect arbitrary faint streaking objects, large groups of pixels must be considered together, so that the noise largely cancels out. If the variance of the local background of the image is σ^2 , then the variance of the mean of n background pixels is σ^2/n due to the law of large numbers. This fact allows the detections of streaked objects with much lower SNRs than is possible with traditional image processing techniques that apply thresholds to individual pixels, even with other filtering tricks. A detection can be made when the mean of the data numbers DN_i for a group of n pixels exceeds a threshold based on the local noise level σ .

$$\frac{1}{n} \sum_{i=0}^n DN_i \geq B + T \frac{\sigma}{\sqrt{n}} \quad (1)$$

where B is the local background level and T would define the threshold in terms of the number of standard deviations of the mean of the group. If the background of

the image is subtracted, this expression can be simplified for convenience.

$$\sum_{i=0}^n DN_i \geq T\sigma\sqrt{n} \quad (2)$$

The challenge of doing this efficiently is that there are too many possible combinations of neighboring pixels (that is, streaks with different orientations and lengths) to consider each combination separately. To do so comprehensively would require an $\mathcal{O}(n^2)$ algorithm, where n is the number of pixels in an image, which would not be feasible for processing large images (4096x4096 px in POGS case) at the same rate that the images can be produced. The strategy implemented to mitigate this problem is to take advantage of the simple geometry of streaks and the fact that pixels in streaks that potentially overlap should not have to be summed up multiple times. That is, if one wishes to test if some group of pixels numbered $\{1, \dots, n\}$ exceeds the desired threshold to decide if they make a detection, there is no need to sum up the same pixels again when testing a larger group of pixels numbered $\{1, \dots, n+1\}$, or an overlapping group like $\{2, \dots, n+1\}$. Accomplishing this is somewhat memory intensive in the pursuit of ”saving” already completed work, but all information can be stored in arrays adding up to the same size as the original image.

First, the background must be accurately estimated and subtracted from the image, such that the expected value of any pixel receiving no star or object signal is zero, and pixels with data numbers below the background level take negative values. The background is estimated iteratively, beginning with the mean value of all pixels in the image. In each iteration, pixels are added to a mask if they are likely to have received signals from stars or objects, and an improved background estimate is calculated from the pixels not in the mask. The background is modeled as a paraboloid. Two-dimensional polynomial coefficients are estimated by a linear least squares procedure. The iteration stops if no new pixels are added to the mask.

Then the image is divided up by lines made at some chosen angle and spaced with some desired width, which should be on the order of the width of the PSF (i.e. the expected width of streaking objects), as shown in Figure 1.

Pixels between each of the lines are grouped together as shown in Figure 2 and cumulative sums are computed along each of those groups, as illustrated in Figure 3.

Then the sums of different groups of pixels along each of those lines may be computed by a single subtraction of two values in the cumulative sum. This reduces the number of operations needed for each ”test,” but a comprehensive check for all combinations would still be $\mathcal{O}(n^2)$. However, if one chooses to test for only streaks of a sensible, finite set of lengths, virtually the same results may be achieved. We propose choosing to test for streaks of exponentially increasing numbers of pixels (10, 20, 40,

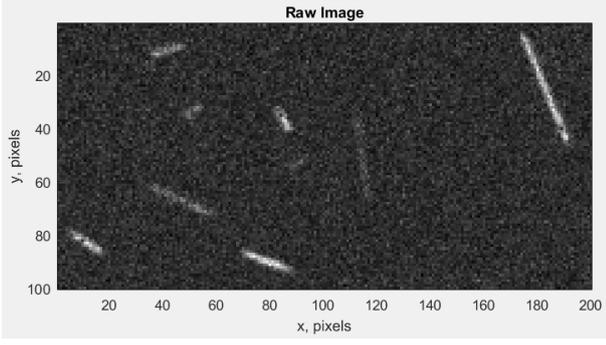


Figure 1. A small artificially generated sample image with several streaks of various sizes and orientations is shown.

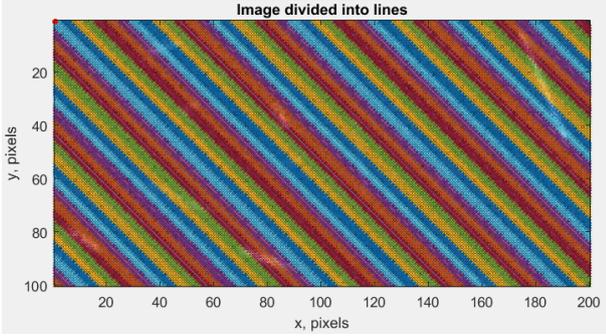


Figure 2. The same image shown in Figure 1 is overlaid with colored markers showing how pixels are grouped into lines with a width of three pixels.

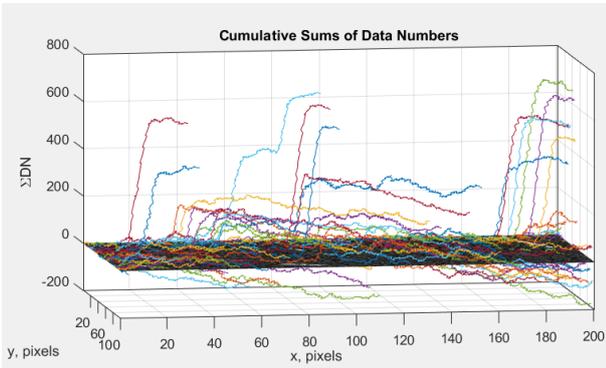


Figure 3. The cumulative sum of each of the lines from Figure 2 is plotted in the third dimension.

80, 160, 320, etc.), essentially reducing the search for detectable objects in an image to $\mathcal{O}(n \log n)$, while maintaining nearly optimal sensitivity to detect streaks near the selected threshold T . This process should be repeated for a variety of angles between 0° and 180° to increase the chances that the cumulative sums will overlap a significant portion of pixels in any detectable streaks in an image.

Once all detections are made, the actual streak parameters (position, length, orientation) are estimated with a quasi-Newton iterative algorithm that fits a mathematical model of a streak to the data numbers. Streaks are modeled as a convolution between a Gaussian point spread function and a 2-dimensional rectangular window as proposed by [9, 3]. Using the same notation as in those papers, the PSF is given by:

$$\Pi(\mathbf{H}) = \exp\left(-\frac{1}{2} \frac{|\mathbf{H}|^2}{\sigma_\Pi^2}\right) \quad (3)$$

where \mathbf{H} is a location relative to the center of the PSF and σ_Π is the width of the PSF (in pixel widths).

$$l(\mathbf{H}) = l_0 \text{rect}\left(\frac{\mathbf{H}_x}{L}\right) \exp\left(-\frac{1}{2} \frac{\mathbf{H}_y^2}{\sigma_y^2}\right) \quad (4)$$

where l_0 is the surface brightness of the streak (luminosity per unit area), L is the length of the streak (in pixel widths), and \mathbf{H}_x and \mathbf{H}_y are defined as:

$$\mathbf{H}_x = \cos \phi_0 (x - x_0) - \sin \phi_0 (y - y_0) \quad (5)$$

$$\mathbf{H}_y = \sin \phi_0 (x - x_0) + \cos \phi_0 (y - y_0) \quad (6)$$

where x and y are any location in the image. The intensity of the streak at $[x, y]$ is then:

$$m = \frac{l_0}{2} \left[\text{Erf}\left(\frac{L - 2\mathbf{H}_x}{2\sqrt{2}\sigma_\Pi}\right) + \text{Erf}\left(\frac{L + 2\mathbf{H}_x}{2\sqrt{2}\sigma_\Pi}\right) \right] \exp\left(-\frac{1}{2} \frac{\mathbf{H}_y^2}{\sigma_\Pi^2 + \sigma_y^2}\right) \frac{\sqrt{2\pi\sigma_y^2}}{\sqrt{2\pi(\sigma_\Pi^2 + \sigma_y^2)}}$$

where σ_y is the width of the rectangular window function (in pixel widths). Ideally, the PSF would be convolved with a one-dimensional line, so σ_y may be an arbitrarily small number. The width of the PSF then determines the width of the streak and $\sigma_y^2 \ll \sigma_\Pi^2$.

In sidereally tracked images, stars will all have very short lengths and random angles, and orbiting objects will appear as long streaks. In object-tracked images, objects

will have short lengths and stars will all have nearly the same length and orientation. Inevitably some stars will be badly estimated due to their proximity to other stars or their locations at the edge of the image.

2.2. Pattern Matching and Plate-Solving

Streaked stars with visual magnitudes greater than 12 have been identified in images from the POGS telescope, and depending on the location in the sky and the exposure time, hundreds of stars can be found in a single image. We seek to maximize the number of stars used for determining the pointing direction of the telescope since the pointing error is expected to decrease as the number of vectors used in plate-solving increases. As such, it is inconvenient to implement restrictions in the matching algorithm, such as requiring stars to be near the center of the image or that the closest star to the central star in a pattern is successfully located, as is the case with Padgett and Kreutz-Delgado's method [7]. The simple solution is to implement a pattern-matching algorithm similar to the grid algorithm, but in polar coordinates. Patterns in polar coordinates can be convolved with each other and matched with any orientation, as long as the central stars of each pattern are the same.

Patterns are generated from detected objects in an image that have been positively identified as stars that are likely to be successfully matched. Such stars can be distinguished by all having similar streak lengths and orientations relative to the image axes. Many stars may be successfully detected, but if the streak properties are not close to the most common lengths and orientations found throughout the image, they are ignored. This can result from stars streaking into or out of the image at the edges (leaving only part of the signal inside the image), or overlapping with other stars and getting estimated as a single streak. From the list of nearly uniform streaks, the brightest one is chosen to be the center of the pattern. The second-brightest streak is chosen to define the primary axis of the polar coordinate grid, but the choice of which star gets used to orient the pattern is not strictly important. The brighter stars are more likely to be in the catalog, however, and matching is easier if the axis star is in both the reference catalog and the image. The central star of a pattern generated from an image *must* be in the reference catalog or no match can be made.

Spherical triangle properties are used to place each star in the pattern. That is, the angles between the central star and all other stars are used as radii in the polar grid, and the angles are computed between the planes containing each pair of vectors (the central star and the axis star, and the central star and each other star). The interior angles of planar triangles do not work well to place points in the pattern, especially near 180° away from the axis star and for images with large fields of view. Care must be taken for stars beyond 180° from the axis star as interior angles of spherical triangles are conventionally less than 180° . In other words, the interior angle may be clockwise or

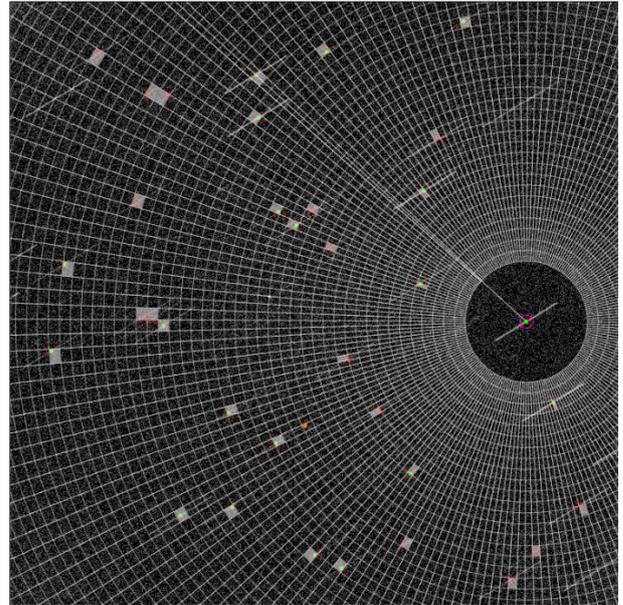


Figure 4. An image taken by the POGS telescope is shown with a pattern populated by 34 detected stars from the image. Red X's show the centers of all detected stars, and the polar axis of the pattern is shown radiating from the central star.

counterclockwise from the axis star, and all angles in the pattern should be measured counterclockwise between 0° and 360° . A simple way to test is to subtract the unit vector of the central star from every other star unit vector to get the relative vectors (which would be the sides of planar triangles) and take the cross product of the axis star's relative vector with every other relative vector. If the result points in the same direction as the central star's unit vector (i.e. the dot product is positive), the interior angle is counterclockwise.

To generate a pattern, radial and angular bins are selected. The size of the pattern and number of bins may be arbitrarily chosen depending on how many stars are expected to be in each pattern. The POGS telescope has a 0.82488° square field of view and the reference star catalog used to make matches is the Tycho-2 catalog [4] with 2.5 million stars. For these conditions, 50 radial bins between 0.1° and 0.8° and 200 angular bins produced good results. The pattern is represented as a 50×200 logical array in Matlab, and stars are placed into the appropriate bin by changing the corresponding entry from 0 to 1. An example of such a pattern is shown in Figure 4.

To match with the reference catalog, a catalog pattern must be generated. Since the approximate pointing direction of the telescope is known a priori, only stars close to that direction need to be considered. The matching algorithm runs through the selected list of catalog stars in order of increasing visual magnitude (decreasing brightness) and generates a pattern with each one as the central star. The patterns are convolved with each other to see how well the patterns match. Visually, this is analogous

to placing one pattern over another and rotating it through each of the angular bins. At each angle, the number of filled bins that overlap is counted and used as a score. If the maximum score between any two patterns exceeds some threshold, then the patterns are said to match each other at the angle where the maximum score was found.

Choosing the threshold is also arbitrary. If large bins are used, random overlaps are expected more often in patterns that shouldn't match. But if a pattern is made with too many small bins, uncertainty in star positions from the image may cause them to be placed in the wrong bins, especially at small radii. However, this method is robust and works with a range of pattern sizes bin widths, as the difference between matching scores and non-matching scores is typically obvious. Defining a threshold as a fraction of the number of matchable stars in the image pattern (such as 30%) works well for large numbers of stars on processed POGS images. Figure 5 visually depicts the pattern from Figure 4 being compared to the catalog. At any angle other than the correct matching angle, three or fewer bins from both patterns overlap. At the correct angle, 20 bins overlap.

Once a pattern is matched, the stars in the pattern must be identified. In the original grid algorithm [7] and the modified polar grid algorithm [5], only the central star in each pattern is matched. To match multiple stars, a pattern is made for each one to be matched in the image. Instead, we propose the radius and angle information are used to match each star in the pattern. To do this, the star that defines the polar axis must be the same in both the image pattern and the catalog pattern. The procedure to match the patterns already indicates the correct relative orientations of the patterns to within one angular bin width. The radii to stars within that angle in each pattern can be compared to find two that match.

Once a star from both patterns is positively identified, the angles to all other stars in both patterns must be defined relative to that star, simply by subtracting them.

$$\vec{\alpha}_{i,cat} = \vec{\alpha}_{i,cat} - \alpha_{0,cat} \quad i = 1, \dots, m \quad (7)$$

$$\vec{\alpha}_{i,img} = \vec{\alpha}_{i,img} - \alpha_{0,img} \quad i = 1, \dots, n \quad (8)$$

where $\vec{\alpha}_{img}$ is a vector of m angles for stars in the image pattern and $\alpha_{0,img}$ is the angle of star that has been chosen to define the polar axis in the image pattern, in degrees. $\vec{\alpha}_{cat}$ is a vector of n angles for stars in the catalog pattern and $\alpha_{0,cat}$ is the angle of the polar axis star in the catalog pattern. Then two $m \times n$ arrays are computed that measure the radial and angular errors between each pair of stars in both patterns.

$$\Delta \mathbf{r}_{i,j} = \vec{r}_{i,img} - \vec{r}_{j,cat} \quad (9)$$

$$\Delta \alpha_{i,j} = [(180^\circ + \vec{\alpha}_{i,img} - \vec{\alpha}_{j,cat}) \% 360^\circ] - 180^\circ \quad (10)$$

where $\vec{r}_{i,img}$ and $\vec{r}_{j,cat}$ are vectors containing the radii for the m stars in the image pattern and n stars in the catalog pattern, respectively. " $\%$ " is the modulo operator. The modulo operation keeps the resulting angle differences in the interval $(-180^\circ, +180^\circ]$, since "large" differences of close to 360° should in fact be considered small negative differences. The angular errors are then scaled by the appropriate radius.

$$\Delta \alpha \mathbf{r}_{i,j} = \vec{r}_{i,img} \Delta \alpha_{i,j} \quad (11)$$

Lastly, the approximate combined squared differences in both (radial and angular) directions are computed.

$$\Delta_{i,j}^2 = (\Delta \mathbf{r}_{i,j})^2 + (\Delta \alpha \mathbf{r}_{i,j})^2 \quad (12)$$

The index of the minimum value in the i^{th} row corresponds to the catalog star that is the closest match to the i^{th} image star. Some stars will not have a match, so a maximum cutoff value is selected to filter bad pairings. A sensible cutoff depends on the size of the FOV and how accurately stars can be located in images.

Image star vectors and catalog star vectors that have been successfully matched are passed as inputs to Shuster's popular QUEST algorithm [10] to determine the orientation of the field of view at the midpoint of the exposure.

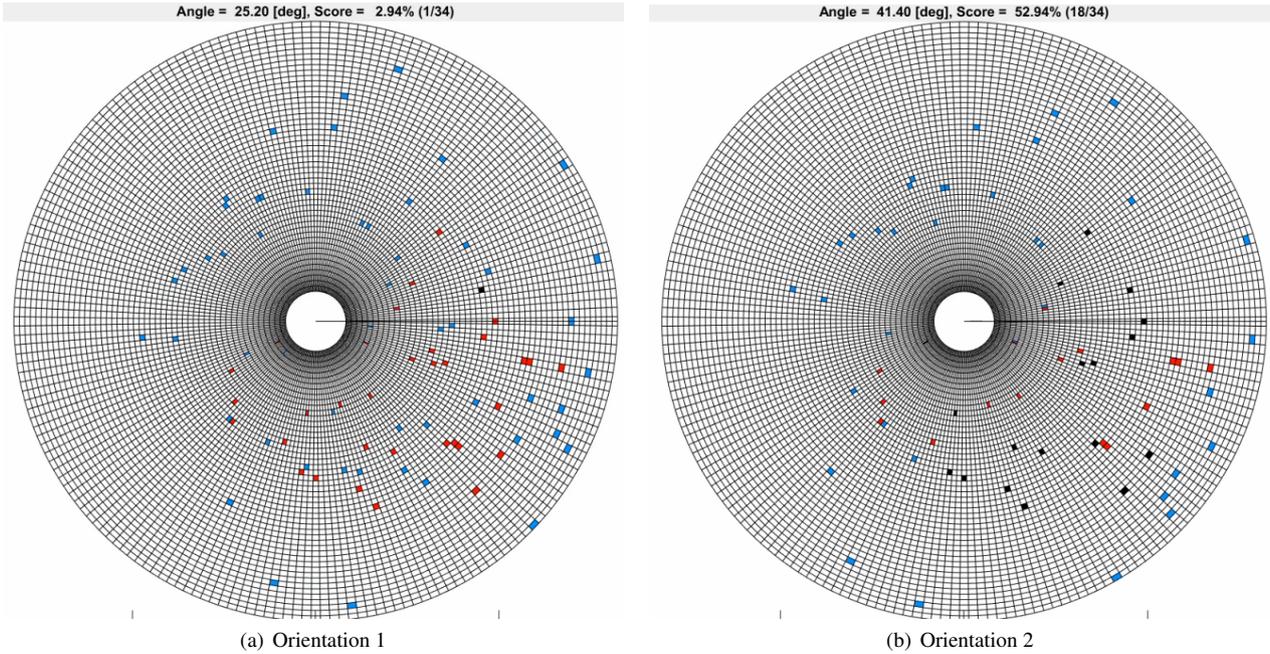


Figure 5. 5(a): A pattern generated from an the Tycho-2 catalog (blue bins) is overlaid on the image pattern from Figure 4 (red bins) at an angle where the patterns do not match. 5(b): The same catalog pattern is rotated 16.2° further to match the image pattern. Overlapping bins are filled in black.

3. RESULTS

These techniques were tested on a variety of images taken by the POGS telescope. For this paper, programs written in Matlab were used on a series of 15 images tracking GPS satellite PRN-31 (NORAD-ID 29486) in June of 2020. Information about detected streaks including 176 successfully matched stars across the 15 images was collected. For comparison, the standard method of applying thresholds to pixels individually was also performed with a 2×2 pixel spatial filter. Two example comparisons are shown in Figures 6 and 7.

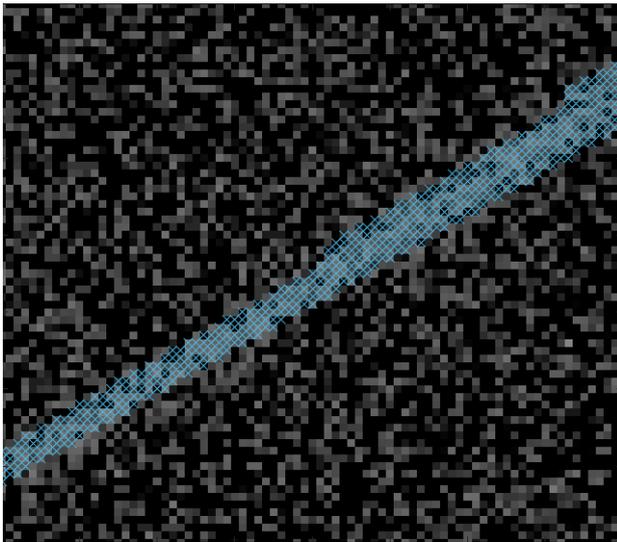
In Figure 6, all pixels in the streak were successfully detected using the cumulative sum method, while the pixel thresholding method caused the streak to get segmented into several separate groups. Some of the groups appear to be touching, but this is only the result of a "buffer region" being added to the edges to capture the fainter parts of a PSF or streak that would not be expected to pass the threshold, but would still contain a part of the received signal. Segmentation like this is a common problem as a result of atmospheric effects and inherent noise causing signals received on the ground to vary over time. Streaks in real images do not have a perfectly consistent brightness throughout, so applying individual pixel thresholds fails to capture larger trends of pixels slightly brighter than the background.

Figure 7 shows another case where barely any of the streaks' pixels are correctly identified with the individual threshold method, as well as a false positive detection in the lower left. The cumulative sum method successfully

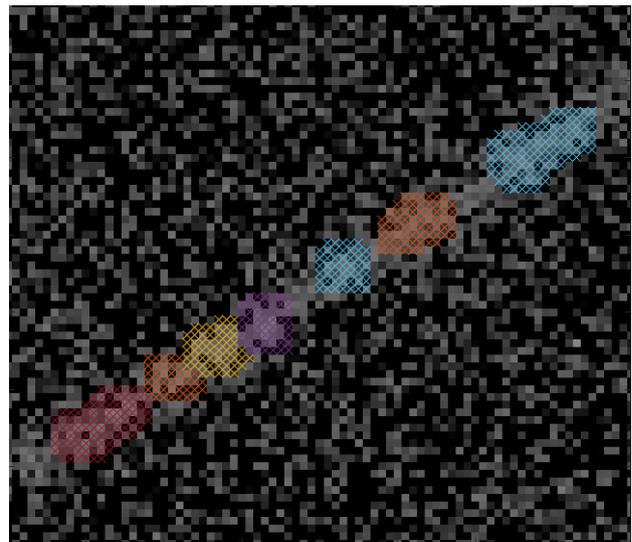
captures both streaks and groups them properly. It is clear from the yellow group that the angle at which pixels are divided into lines does not align exactly with the angle of the streaks, resulting in a "step" shape to the group of detected pixels. Additionally, some noise or imperfect background estimation has caused pixels beyond the end of the purple streak and above the yellow streak to be included in the detections. However, after iteratively fitting the mathematical streak model to the data numbers of the detected pixels, good estimates of the lengths and angles are calculated, as shown by the red lines. When streaks touch the edge of the image, signal from another object, or each other, even the iterative fitting procedure may fail to accurately estimate their properties, and no attempt to match them with the catalog is made.

The matching algorithm successfully identified an average of 11.7 stars per image in the Tycho-2 catalog. The 20 stars that were identified in the image from Figure 4 are labeled in Figure 8.

To evaluate how well the centers of the streaks are located in these images, the unit vectors computed from the 176 image streaks that were positively identified as stars were compared to the matched vectors from the Tycho-2 catalog. The unit vectors from the catalog were rotated into the image axes using the solution from QUEST. The angles between them and the corresponding image unit vectors were found to have an average of 0.9 arcseconds of error with a standard deviation of 0.6 arcseconds. Matched stars had visual magnitudes as high as 12.4, as shown in Figure 3. The Tycho-2 catalog is reported to be 99% complete at magnitude 11 and only 90% complete at magnitudes of 11.5 [4]. It is likely that fainter stars

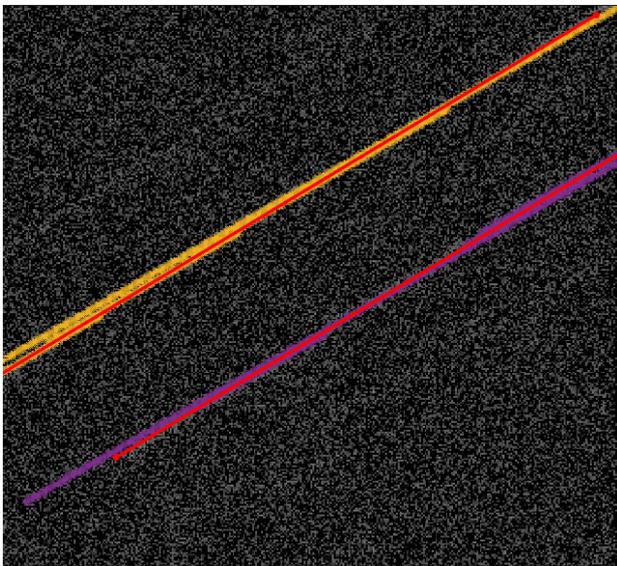


(a) Cumulative Sums

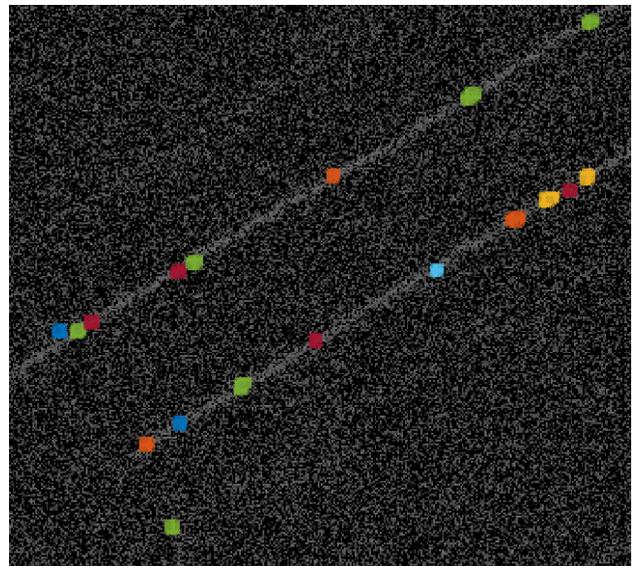


(b) Pixel Thresholds

Figure 6. 6(a): A close zoom to a part of an image from Figure 4 containing a streak is shown, with pixels detected by cumulative sums marked in blue. 6(b): The same part of the same image shows detections by applying a threshold to each pixel individually. Each group is marked with its own color for clarity.



(a) Cumulative Sums



(b) Pixel Thresholds

Figure 7. 7(a): Another part of the same image shows two streaks detected by cumulative sums in purple and yellow. The endpoints of the fitted streaks are marked by red dots and connected lines. 7(b): The same pair of streaks and the threshold detections are shown.

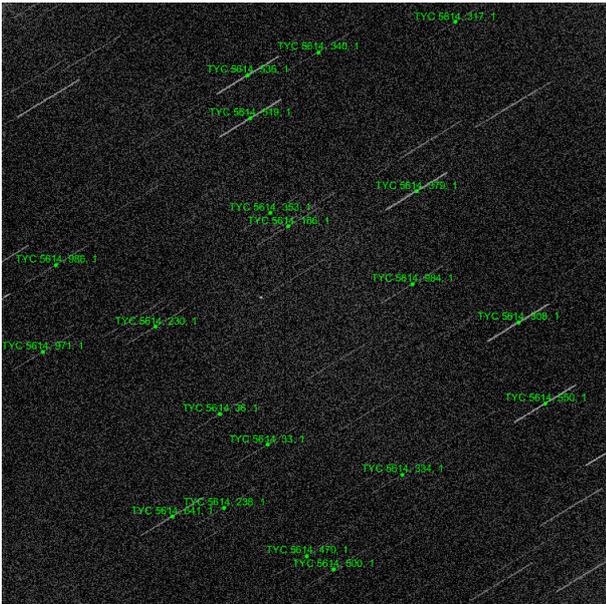


Figure 8. The image from Figure 4 is labeled with the stars that were identified in the Tycho-2 catalog.

were found (and correctly identified as stars) that are not in the Tycho-2 catalog.

4. CONCLUSIONS

The cumulative sum method for streak detection is robust against fragmenting streaks, and does well to detect all pixels in a streak, unlike commonly used thresholding of individual pixels. Long exposure times make overlapping streaks more likely, and current methodology does not allow such detections to be accurately estimated. Isolated streaks that are completely detected have had magnitudes as high as 12.4 and the iterative fitting procedure locates them to an average error of 0.9 arcseconds. If the errors are unbiased and attributable to image noise, the attitude estimates are expected to be even more accurate than the average error in the star vectors.

While overlapping streaks can cause even the brightest stars in the image to be thrown out before the matching procedure, the polar grid method works well as long as the central star in the image pattern exists in the reference catalog. The polar grid method also allows for only part of the pattern to be visible in the image while still guaranteeing it will be matchable with the reference catalog. It is possible to quickly match large numbers of stars (sometimes hundreds) in a single image. The ability to detect, locate, and match large numbers of stars is important to maximize the accuracy of the attitude solution.

REFERENCES

1. COLE, C., AND CRASSIDIS, J. Fast star pattern

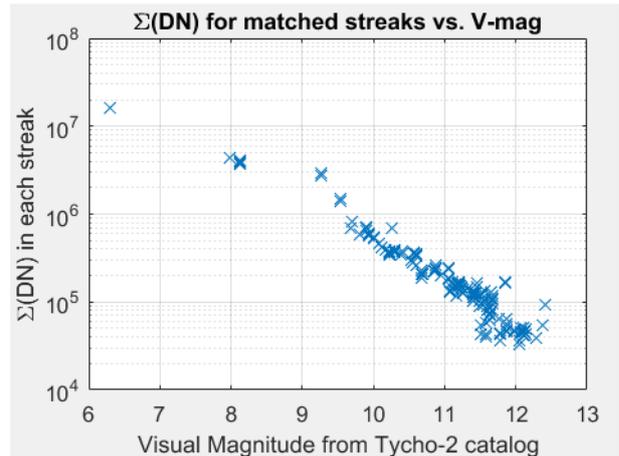


Figure 9. The sum of data numbers over pixels belonging to each matched streak in a series of 15 images is plotted against the corresponding visual magnitude from the Tycho-2 catalog.

recognition using spherical triangles. In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit* (2004), p. 5389.

2. COLE, C. L., AND CRASSIDIS, J. L. Fast star-pattern recognition using planar triangles. *Journal of guidance, control, and dynamics* 29, 1 (2006), 64–71.
3. DAWSON, W. A., SCHNEIDER, M. D., AND KAMATH, C. Blind detection of ultra-faint streaks with a maximum likelihood method. *arXiv preprint arXiv:1609.07158* (2016).
4. HOG, E., FABRICIUS, C., MAKAROV, V. V., URBAN, S., CORBIN, T., WYCOFF, G., BASTIAN, U., SCHWEKENDIEK, P., AND WICENEC, A. The tycho-2 catalogue of the 2.5 million brightest stars. Tech. rep., NAVAL OBSERVATORY WASHINGTON DC, 2000.
5. LEE, H., AND BANG, H. Star pattern identification technique by modified grid algorithm. *IEEE Transactions on Aerospace and Electronic Systems* 43, 3 (2007), 1112–1116.
6. MORTARI, D. Search-less algorithm for star pattern recognition. *The Journal of the Astronautical Sciences* 45, 2 (1997), 179–194.
7. PADGETT, C., AND KREUTZ-DELGATO, K. A grid algorithm for autonomous star identification. *IEEE Transactions on Aerospace and Electronic Systems* 33, 1 (1997), 202–213.
8. SCHILDKNECHT, T., HUGENTOBLER, U., VERDUN, A., AND BEUTLER, G. Ccd algorithms for space debris detection. *ESA/ESOC contract No. 10623/93/D/IM Final Report* (1995).
9. SCHNEIDER, M. D., AND DAWSON, W. A. Synthesis of disparate optical imaging data for space domain awareness. *arXiv preprint arXiv:1609.07157* (2016).

10. SHUSTER, M., AND OH, S. Three-axis attitude determination from vector observations. *Journal of Guidance and Control* 4 (1981), 70–77.
11. SPRATLING, B. B., AND MORTARI, D. A survey on star identification algorithms. *Algorithms* 2, 1 (2009), 93–107.
12. VANANTI, A., SCHILD, K., AND SCHILDKNECHT, T. Streak detection algorithm for space debris detection on optical images. In *Proceedings of AMOS Conference* (2015).
13. WEI, X., WEN, D., SONG, Z., XI, J., ZHANG, W., LIU, G., AND LI, Z. A star identification algorithm based on radial and dynamic cyclic features of star pattern. *Advances in Space Research* 63, 7 (2019), 2245–2259.
14. ZHANG, G., WEI, X., AND JIANG, J. Full-sky autonomous star identification based on radial and cyclic features of star pattern. *Image and vision computing* 26, 7 (2008), 891–897.
15. ZIMMER, P. C., ACKERMANN, M. R., AND MCGRAW, J. T. Gpu-accelerated faint streak detection for uncued surveillance of leo. In *Proceedings of the 2013 AMOS Technical Conference* (2013).