



LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

Autonomous Sub-Pixel Satellite Track Endpoint Determination for Space Based Images

L. M. Simms

March 9, 2011

Applied Optics

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

Autonomous Sub-Pixel Satellite Track Endpoint Determination for Space Based Images

Lance M. Simms^{1,*}

Lawrence Livermore National Laboratory, Livermore, CA, 94550 USA

**Corresponding author: simms8@llnl.gov*

An algorithm for determining satellite track endpoints with sub-pixel resolution in spaced-based images is presented. The algorithm allows for significant curvature in the imaged track due to rotation of the spacecraft capturing the image. The motivation behind the subpixel endpoint determination is first presented, followed by a description of the methodology used. Results from running the algorithm on real ground-based and simulated spaced-based images are shown to highlight its effectiveness. © 2011 Optical Society of America

1. Introduction

The issue of autonomously detecting satellite and airplane tracks in images is by no means a new one. For decades, these tracks have been nothing more than a nuisance for astronomers—foreground artifacts that must be disposed of in the preprocessing of data—and several methods for identifying and removing them have been discussed in the literature. For instance, the RAST algorithm [1] removes satellite streaks directly from images using a geometric approach that assumes the tracks are straight lines and Storkey et al. [2] use the RANSAC algorithm to allow for post-processing removal of curved tracks and scratches as well.

While these streaks may be a source of noise in the field of astronomy, for applications such as the Space Surveillance Network (SSN) they are the signal. A track from a satellite or piece of debris, along with time-stamp information, allows the SSN to make an equatorial angles-only determination of its orbit. One can conceive of several ways of obtaining the time-stamp information, but the most straightforward approach is to measure the start and end times of an exposure and extract the endpoints of the imaged track(s).

The precision of such a measurement, of course, is dependent on how well one can determine the track endpoints on the detector. As Earl notes, the error in detecting the endpoints may

very well dominate the other sources of error in the measurement [3]. Increasing the detector resolution (number of pixels per unit area) to mitigate this error is not a viable option because doing so decreases the dwell time per pixel of the target, effectively lowering its signal to noise. But even with low resolution detectors, sub-pixel information is still available since the time spent by the satellite “in” the pixel translates to intensity, so all hope is not lost.

In fact, several methods to obtain sub-pixel track endpoints are available. For instance, Levesque presents an algorithm for accurate endpoint detection that has been successfully used on images obtained with the Canadian Automated Small Telescope for Orbital Research (CASTOR) system [4]. However, the problem with these methods is they generally make the assumptions that **1**) the track is *straight* and **2**) previously obtained orbital information is available to predict the appearance of the streak in the newly acquired image.

The motivation behind the method that will be discussed in this paper is a mission called Space-based Telescopes for Actionable Refinement of Ephemeris (STARE) for which neither of these assumptions is valid [5]. The purpose of STARE is to refine orbital information for satellites and debris by directly imaging them with CMOS imagers on-board two separate Cubesats. The images acquired by a given sensor will be run through an algorithm in the on-board microprocessor that is tasked with extracting star and track endpoint coordinates and sending them to the ground (without the accompanying image). Since the attitude of the STARE satellites will not be precisely controlled, the telescopes may be rotating about the pointing axis. And uncertainty in the initial orbits means the location of the tracks on the imager will not be well known. The STARE algorithm must therefore deliver sub-pixel endpoint determination for tracks with arbitrary curvature and location.

It should be emphasized that the algorithm is not concerned with detection of faint streaks, but rather high fidelity endpoint determination for streaks with ample Signal-to-Noise Ratio (SNR). Also, to avoid confusion while describing the algorithm in the following sections, the term *satellite* will be reserved for the STARE Cubesat. The imaged debris or satellite will be referred to as the *target*.

2. Curved Target Tracks in STARE Images

Any movement of the STARE satellite during an exposure is obviously unwanted, as it will effectively decrease the dwell time per pixel of the target. But rotation of the satellite about the two axes perpendicular to the telescope pointing is of less concern because it simply adds to the transverse velocity component of the target and causes the stars to streak in a uniform manner across the detector.¹ It will not produce curvature in the streak left by the target.

¹ Note that a simplification has been made by approximating the path of the target as a straight line during the exposure, which it is not.

Rotation about the pointing axis, on the other hand, could potentially induce significant curvature. If the satellite has a rotational velocity of $\dot{\theta}$ about the pointing axis, which will be taken as z , and the target has velocity components (v_x, v_y, v_z) and coordinates of

$$x = x_o + v_x t, \quad y = y_o + v_y t, \quad z = z_o + v_z t, \quad (1)$$

with respect to the satellite center of mass, then the location of the target in the detector coordinate system is given by

$$x' = (x'_o + v'_x t) \cos(\dot{\theta}t) + (y'_o + v'_y t) \sin(\dot{\theta}t), \quad y' = -(x'_o + v'_x t) \sin(\dot{\theta}t) + (y'_o + v'_y t) \cos(\dot{\theta}t), \quad (2)$$

where the primes represent the mapping of object space to pixel space and rotation of the satellite about the x and y axis has been folded into the components v_x and v_y .

One can gain an appreciation for the form of Equation 2 by considering that for the case of $x_o = y_o = 0$, it is the parametric representation of a spiral. Telescope angular velocities above $0.1^\circ/\text{s}$ are not anticipated, so a spiral pattern should never be observed in STARE images. But $\dot{\theta} = 0.1^\circ/\text{s}$ is large enough to make a Hough Transform ineffective for basic detection and create an error as large as two pixels for a track that extends all the way across the image if a global linear fit is used.

Fortunately, fitting the entire track is not necessary. As long as the parameters $\dot{\theta}_x$, $\dot{\theta}_y$, and $\dot{\theta}_z$ are known reasonably well², the track endpoints (x'_o, y'_o) , (y'_f, y'_f) are sufficient to refine the orbit of the target. The primary intent of the STARE algorithm is to find these coordinates.

3. STARE Endpoint Determination Algorithm

The following subsections follow the numbering in Figure 1, which gives an overview of the STARE algorithm.

3.A. Image Correction

Before the images are searched for stars and tracks, they must first be cleaned. Because the STARE algorithm identifies stars and tracks as a contiguous set of pixels above a noise threshold, T , pre-processing of the data is crucial to its success. The basic steps of the image correction, shown in box **1** of Figure 1, are as follows:

1. Sky Image or Background Subtraction

In the case of the STARE mission, 10 raw images slightly offset from each other will be median filtered to produce a sky image. Subtracting this sky image from a raw image very accurately removes both dark current and sky background. If a sky image is not available, modal subtraction or other methods of background subtraction can be used.

² We should have this information from calibration data taken before the observation.

2. Bad Pixel Masking

Bad pixels are problematic for thresholding. These pixels can easily be mapped during routine calibration of the detector and stored as a mask in non-volatile memory. They are zeroed in each of the background subtracted images so they do not contaminate filtering in the next step.

3. Low Pass Filter

The corrected image is smoothed using a Gaussian kernel with a FWHM on the order of two pixels. The smoothing fills in reasonable values for the zeroed pixels and ensures that tracks are contiguous. If the bad pixel density becomes excessive, the kernel can be extended at the expense of increasing the error in endpoint estimation.

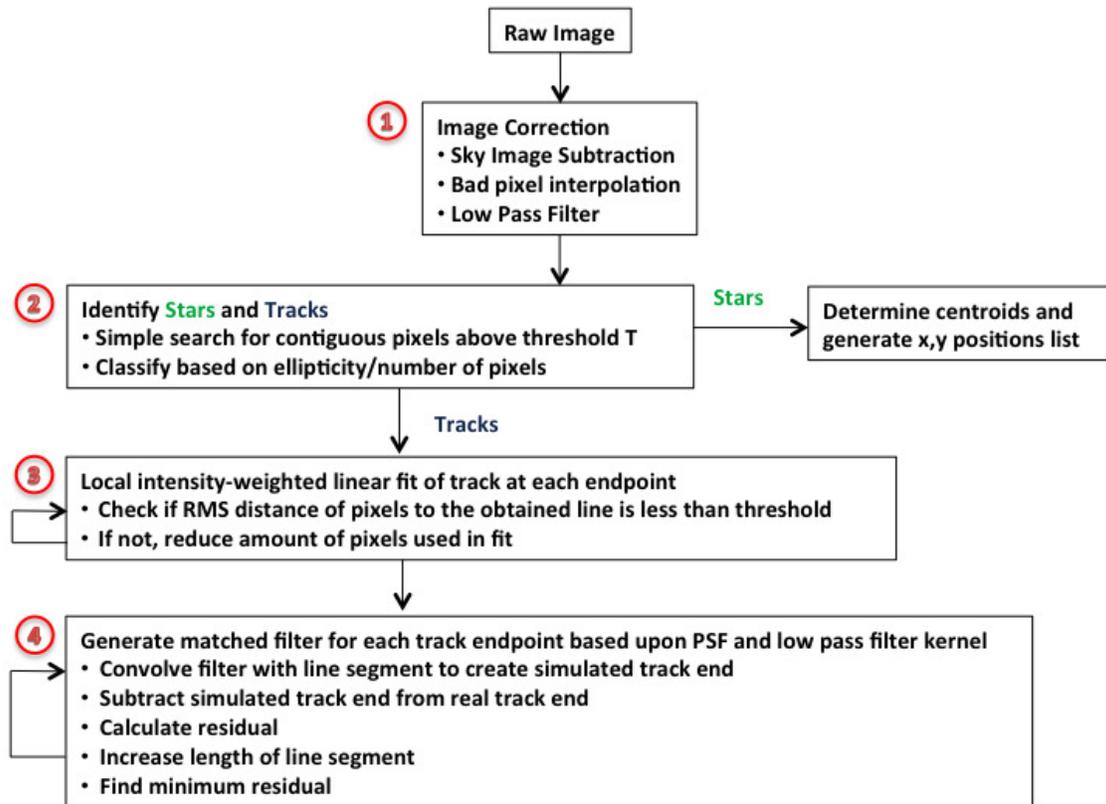


Fig. 1. Flow diagram for the various steps used in the STARE endpoint detection algorithm. Each of the circled numbers corresponds to one of the subsections in this section.

3.B. Object Detection

After the image is corrected, it is searched for contiguous sets of pixels that have a value above T . This step is shown in box **2** of Figure 1. With both real and simulated images, typically $T = 3.5 * RN$, where RN is the read noise of the detector, produces good results. If sky noise or dark current shot noise dominates the read noise, this must be taken into account in setting T .

Once a contiguous set of pixels has been identified, it is characterized as a star, track, or unknown object (such as a delta or Compton scattered worm) based upon its ellipticity (e) and the number of pixels (N) it contains. These values are dependent on the optical system and detector used, but for STARE a cut of $e > 0.8$ and $N > 20$ should effectively identify all real tracks. A perfectly straight track should have $e = 1$; the margin $e = 0.8 - 1.0$ allows for curvature and the possibility of overlapping stars or cosmic rays. The chance of a muon hit producing a track greater than 20 pixels long is extremely low.

Confusion of cosmic rays and stars could potentially be more troublesome. For instance, the STARE optical system produces a sub-pixel Point Spread Function (PSF), and most stars will actually appear as 1-4 pixel points rather than the nice gaussian profiles encountered in astronomy applications. Based on previous space based measurements, though, a significant amount of 1-4 pixel cosmic ray events are not anticipated in the STARE one second exposures [6,7]. At geomagnetic latitudes below 50° , about 0.706 events per exposure are expected, and above 50° this number may go up to 12. With these rates, an astrometric solution from the list of star centroids is possible even with the contamination.

3.C. Iterative Local Fitting at Track Endpoints (Transverse Degree of Freedom)

The next step, step **3**, is to find the endpoints for each of the tracks identified in step **2** above. As previously mentioned, applying a global linear fit to the track to find its endpoints may result in large errors. But a local linear fit to the track at each endpoint can still help in constraining their possible locations. The question that then arises is how many pixels to use in the fit. If too many are used, the curvature of the track will force the slope toward the global average. If too few are used, the estimate is vulnerable to detector noise, bad pixels, etc.

One might consider using the second derivative as a criterium:

$$\frac{d^2y'}{dx'^2} = \frac{2\dot{\theta}(-v'_x \sin(\dot{\theta}t) + v'_y \cos(\dot{\theta}t)) - \dot{\theta}^2((x'_o + v'_x t) \cos(\dot{\theta}t) + (y'_o + v'_y t) \sin(\dot{\theta}t))}{2\dot{\theta}(-v'_x \cos(\dot{\theta}t) - v'_y \sin(\dot{\theta}t)) - \dot{\theta}^2((x'_o + v'_x t) \sin(\dot{\theta}t) + (y'_o + v'_y t) \cos(\dot{\theta}t))} \quad (3)$$

(note that any change in the angular velocity has been ignored, $\ddot{\theta} = 0$). But this expression requires accurate knowledge of x'_o , y'_o , v'_x , and v'_y , which will not be known.

A solution to the problem is to use an iterative weighted least squares fit to each track endpoint until the root mean square deviation of distance from the included track pixels to

the line is below a certain threshold, σ_D^{max} . Starting with all $N_{pix} = N$ pixels identified in the track, a line is fit using the expression:

$$m = \frac{\sum_{i=0}^{N_{pix}} x'^2 \sum_{i=0}^{N_{pix}} I y' - \sum_{i=0}^{N_{pix}} I x' \sum_{i=0}^{N_{pix}} I x' y'}{N_{pix} \sum_{i=0}^{N_{pix}} I x'^2 - \left(\sum_{i=0}^{N_{pix}} I x'\right)^2}, \quad b = \frac{N_{pix} \sum_{i=0}^{N_{pix}} I x' y' - \sum_{i=0}^{N_{pix}} I x' \sum_{i=0}^{N_{pix}} I y'}{N_{pix} \sum_{i=0}^{N_{pix}} I x'^2 - \left(\sum_{i=0}^{N_{pix}} I x'\right)^2}, \quad (4)$$

where I is the pixel intensity and the indices on x' , y' , and I have been left out for notational convenience. Then the distance of the track points to the line is calculated using

$$D = \frac{I(m x' - y' + b)}{I_{max} \sqrt{m^2 + 1^2}}, \quad (5)$$

where I_{max} is the maximum pixel intensity for the N_{pix} pixels used in the fit. If the RMS of this value, σ_D , is below the threshold σ_D^{max} then the fit is considered valid. If not, n pixels are removed from the end of the track opposite to the one being fit and the above procedure is repeated. Thus, at the j^{th} iteration, the track end will be fit with $N_{pix} = N - n * j$ pixels. A minimum number of pixels to be used in the fit $N_{pix} = N_{min}$ is also incorporated, the value depending on the maximum curvature expected.

The threshold σ_D^{max} and whether intensity weighting is used in Equation 5 will depend on the potential curvature and actual PSF of the system. Figure 2 shows results for a simulated track where $\dot{\theta} = 1.0^\circ/s$ and $\sigma_D^{max} = 0.50$ was used without weighted fitting. The eventual error in endpoint estimation was less than 0.1 pixels in both x and y . One can imagine extreme cases in which the target traces out a path perfectly centered over the dividing boundary between two rows of pixels, but this will be a very rare occurrence.

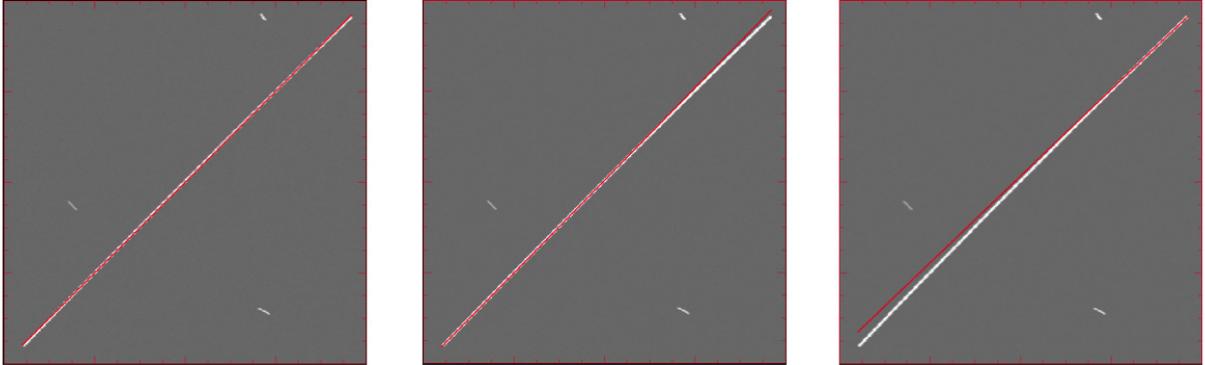


Fig. 2. An example of the local fitting at each endpoint. The left image shows the track fit in red when all pixels were used, the middle when the left 200 pixels were used, and the right when the right 170 pixels were used.

3.D. Matched Filter at Track Endpoints (Longitudinal Degree of Freedom)

Once the track has been fit at each endpoint, the path the target took along the detector near that point is well approximated. What is left is to determine precisely where the target was along this path at the start (or end) of the exposure (step 4). Simply recording the first or last pixel with a value above T will obviously result in errors. Accurately determining the location of the target requires that taking into account the PSF of the optical system and the kernel used in the low pass filter of step 1.

To do this, a Region Of Interest (ROI) around the roughly estimated endpoint that spans $R \times R$ pixels is first considered. An example ROI with $R = 7$ is shown in (a) of Figure 3. The goal is to reproduce this ROI with a simulated one obtained by convolving a line segment with a filter that matches the PSF and kernel described above. The form of the line segment is already known from the fit obtained in step 3. The length of it will indicate exactly where the endpoint is located.

After dividing each simulated pixel into r subpixels, a line segment of length $L = 1/r$ is created at the edge of of the simulated ROI from which the track emerges. The segment is convolved with the filter to produce a track in the simulated ROI, as shown in (b) of Figure 3. The simulated ROI is then subtracted from the real one and the residual is squared. The length of the line segment is increased by $1/r$ and the process is repeated so that after $R * r$ iterations, there will be a set of $R * r$ residuals. The minimum of these, as shown in (d) of Figure 3, indicates where the endpoint is located.

4. Results for Simulated and Real Images

The results from testing the STARE algorithm on real images obtained by ground based telescopes are encouraging. For these images, a median sky frame and bad pixel map could not be obtained, but subtraction of the mode sufficed for image correction. In Figure 4, tracks found in three separate Oceanit images are shown after being analyzed by the algorithm. The ends of the green line segment indicate where the extracted endpoints are located. Although there are no official coordinates for these reported in the Oceanit data, inspection by eye shows that they line up well with the locations expected from the 1.9 pixel FWHM PSF.

Extensive testing on simulated tracks and star fields has also been performed. These tests are especially useful because the measured endpoint can be compared to the true endpoint to determine the accuracy of the algorithm as a function of track length, orientation, brightness, etc. To comprehensively measure the error in the estimated endpoints, a 10 hour run was performed in which 400 images were generated and analyzed. Real star fields were sampled and then tracks with random orientation and length were generated in a number of different brightness intervals. As a proxy for brightness, the quantity of *photons per micron*, which is the x-axis of Figure 5, was used. The reason for this is that a track of a given brightness

will produce varying signal to noise ratios depending on how it is oriented relative to the detector. For instance, if a track is centered over the boundary between a row of pixels, it will produce roughly half the SNR as it would when centered directly over one of the two rows.

On the y-axis of Figure 5 is the total error in the endpoint estimate, $Err = \sqrt{x_{err}^2 + y_{err}^2}$, where x_{err} and y_{err} are simply the difference between the real and measured coordinates. The plot shows that at a level of about 600 photons per micron, the error approaches a near

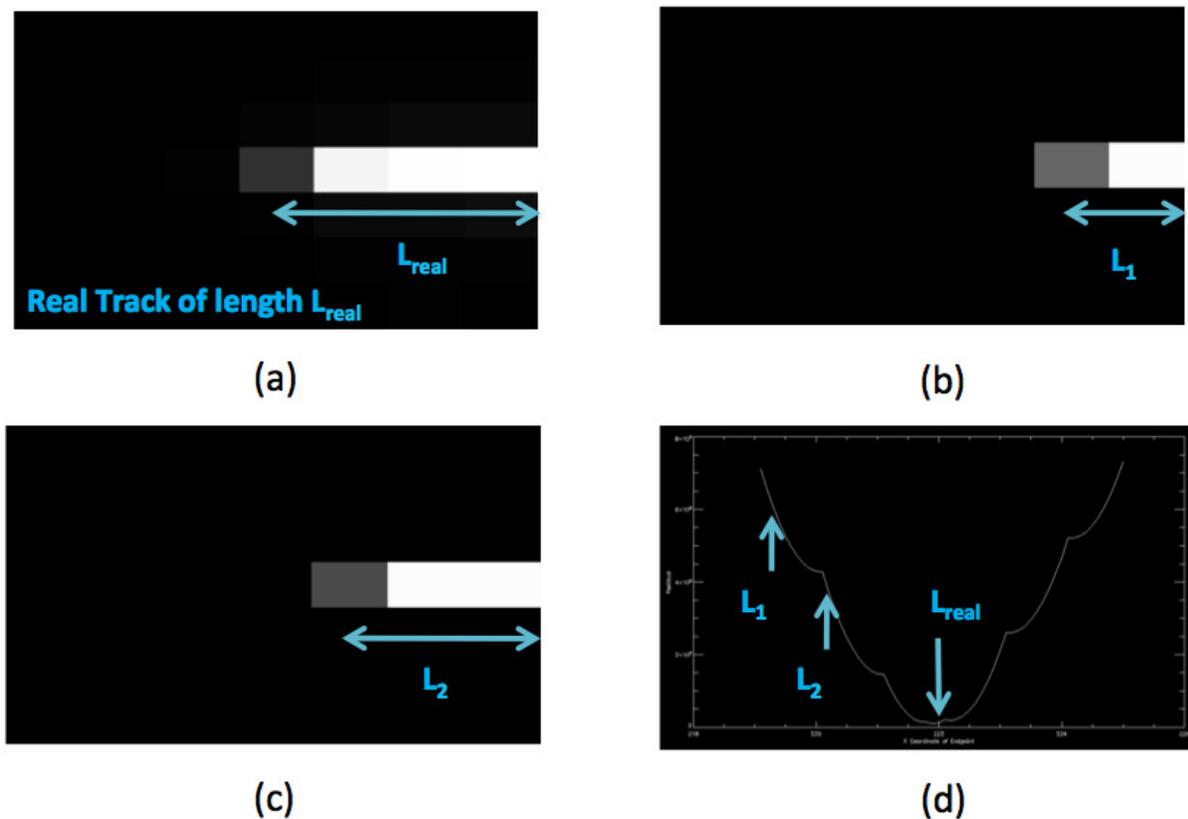


Fig. 3. Illustration of the matched filter process. a) shows an ROI taken from the a corrected raw image. b) shows a simulated ROI, where a line segment of length L_1 has been convolved with a match filter to attempt to reproduce the real track in 1). In c) the length has been extended to L_2 as part of the iterative process. And in d), the entire simulated ROI has been spanned to produce a residual at all $R * r$ grid points. The real track length L_{real} is evident at the minimum of the residual curve.

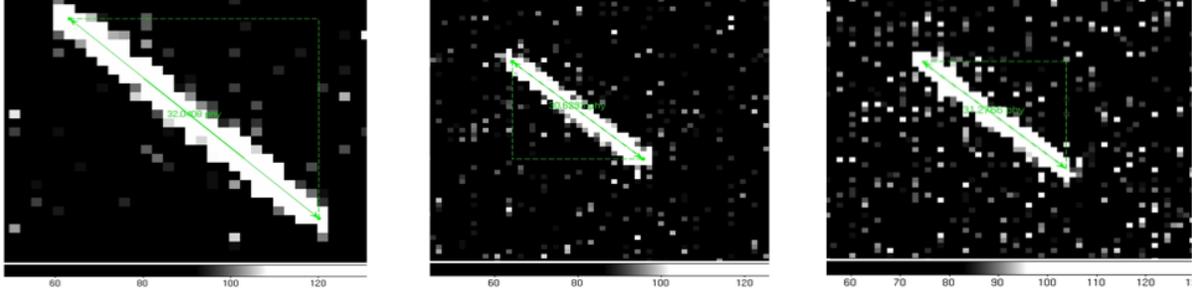


Fig. 4. Endpoint determination for satellite track detected in three separate Oceanit images. While precise endpoint coordinates are not available for comparison as they are in the simulated images, the reported endpoints match up well with what we expect based on the PSF of the system.

constant value of $Err = 0.14$. This is expected from the choice of $r = 10$ for the simulated grid, which should produce an error of roughly 0.1 pixels for each coordinate (the step in length at each iteration is $L = 0.1$ pixels). The value of 600 photons per micron corresponds to a SNR in the range of 6-12, depending on the track orientation. One can see that at a value of 250 photons per micron, which is roughly a SNR of 2-4, the error is slightly larger. But it is still sub-pixel and will serve well for the purpose of orbital refinement.

5. Discussion

The results in the previous section show that the STARE algorithm is capable of extracting accurate track endpoints for both straight and curved tracks. The error of 0.14 pixels obtained for the simulated data can be decreased further by increasing the number of grid points per pixel r at the expense of increased computation time as long as the track has sufficient SNR. The accuracy cannot be improved indefinitely for very bright objects, of course, as it will be limited by the dynamic range of the detector, dwell time per pixel of the target, etc.

One may point out that a disadvantage of the technique is that it requires the track to be a contiguous set of pixels. A large fraction of satellites oscillate in brightness as they cross the sky and their signal may fall below the noise threshold as a result. However, as long as the value N is set low enough, the only implication is that the endpoints for a number of sub-tracks will be reported instead of two (this would also be the case if there is an extensive region of bad pixels the track happens to cross). If the target happens to reach a minima in brightness at the start and end of the exposure, there is no hope of accurately measuring the endpoint anyway.

Another important point to consider is that, although it will not be available in the

STARE mission, a priori knowledge of the target could potentially be used to enhance the performance of the algorithm. A rough estimate of the velocity and position at the exposure start can be used in Equation 3 to help determine the number of pixels used to fit each end of the track directly or help in determining a value for σ_D^{max} . And if the information is accurate enough, it may be possible to generate a matched filter for initially finding the track even for the case of high curvature.

It should also be mentioned that the numbers presented in the previous section are for a very idealized scenario. A number of other errors—GPS measurement errors, timing errors, attitude control uncertainty, etc.—come into play in the game of orbital refinement. The simulations have neglected these. Also, the simulations ignore the low fill factor of the CMOS

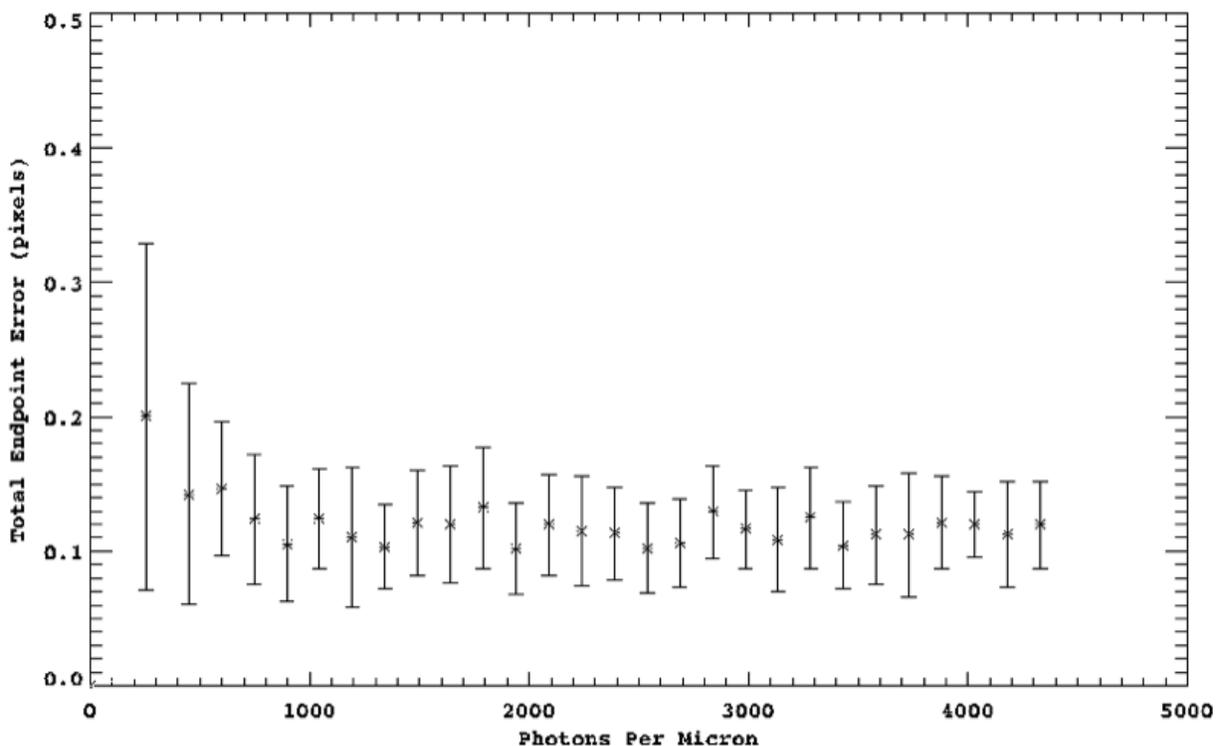


Fig. 5. A plot showing the total endpoint error from a run of 400 tracks of random lengths, orientation, and brightness. The y-axis shows the total endpoint error and the x-axis shows photons per micron, both of which are described in the text. At 250 photons per micron, the SNR ranges from 2-4. At 600 photons per micron, the SNR ranges roughly from 6-12. These values depend on the orientation of the track relative to pixel boundaries.

detector that will be used for the STARE mission. Because the pixels are not sensitive over their entire area, information is lost every time the target spot passes over the pixel boundaries, and this alone can produce 0.3-0.7 pixel errors.³ As long as these systematics remain reasonably well behaved, though, the sub-pixel results provided by the STARE algorithm will allow for orbital refinement.

6. Conclusion

An algorithm for determining the endpoints of satellite and debris tracks in space-based images has been presented. The algorithm is capable of delivering sub-pixel accuracy even for the case of curved tracks resulting from rotation of the imaging spacecraft. The underlying methodology and motivation for the algorithm have been discussed, and results for both real and simulated data showing high quality performance have been presented. Results from real data obtained by the STARE satellites will soon follow.

References

1. H. Ali, C. Lampert, and T. Breuel, "Satellite tracks removal in astronomical images," in "Progress in Pattern Recognition, Image Analysis and Applications," , vol. 4225 of *Lecture Notes in Computer Science* (Springer Berlin / Heidelberg, 2006), pp. 892–901.
2. A. J. Storkey, N. C. Hambly, C. K. I. Williams, and R. G. Mann, "Cleaning sky survey data bases using hough transform and renewal string approaches," *Monthly Notices of the Royal Astronomical Society* **347**, 36–51 (2004).
3. M. A. Earl, "Determining the Range of an Artificial Satellite Using its Observed Trigonometric Parallax," *Journal of the Royal Astronomical Society of Canada* **99** (2005).
4. M. Levesque, "Automatic reacquisition of satellite positions by detecting their expected streaks in astronomical images," in "Advanced Maui Optical and Space Surveillance Technologies Conference," (2009).
5. Simms et al., "Optical payload for the STARE mission," (April 24-28, 2010), Proceedings of SPIE Defense and Security Conference.
6. D. Shaw and P. Hodge, "Cosmic ray rejection in STIS CCD images," Tech. rep. (1998).
7. Alcaraz et al., "Protons in near earth orbit," *Physics Letters B* **472**, 215–226 (2000).

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

³STARE is a pathfinder mission for a future constellation of Cubesats that will carry high quality sensors that will not suffer from this problem.