Prototype of automatic satellite streak detection, identification and initial orbit determination pipeline from optical observation

M. D. Danarianto¹, A. M. Maharani², B. M. Falah³ and F. Rohmah¹

¹ Research Center for Space, National Research and Innovation Agency, Indonesia
² Physics Study Program, Universitas Pendidikan Indonesia, Jl. Setiabudhi 229, Bandung, Indonesia
³ Department of Astronomy, Faculty of Mathematics and Natural Sciences, Institut Teknologi Bandung, Jl. Ganesha No. 10, Bandung, 40132, Indonesia

E-mail: dio.danarianto@lapan.go.id

Abstract. We present a working prototype of automatic satellite streak detection, identification and initial orbit determination pipeline from optical telescope images based on python. The pipeline process calibrated and plate-solved astronomical images in fits format. We utilized Hough transform-based feature detection to detect the light streaks and extract the equatorial position based on the image header information. The streaks were then identified by comparing their center position with the calculated SGP4 based satellite position from available TLE data. The corresponding initial orbit is calculated by utilizing the Laplacian method by interpolating the streak position and time. We test the pipeline by using 85 blind-survey 10 cm telescope images around the celestial equator. The pipeline correctly detects 123 lines with 0.85 sensitivity, automatically identify 92 detected satellites unambiguously and determine the initial orbit of 118 streaks.

1. Introduction

Since the first colonization of space by human-made objects, the number of satellites in space is increasing continuously with rapid change within the past two decade. The recent upward trend in number of satellites for the past five years is powered by the demand of commercial satellites for various human activity. By the beginning of 2021, humans were able to deploy up to 143 satellite in a single rocket launch and by September 2021, there are about 7500 satellite in the orbit with only about 60% of them is still active [1]. These objects should be tracked to maintain information about space environment and related hazard for both in space and ground. In this context, monitor of such an object in high accuracy needs to be conducted periodically to maintain the exact orbital information of these object in space. However, these orbital parameter for both active and inactive satellites and space debris is always changing due to the nature of higher order effects on space such as Earth-Moon perturbation, atmospheric drag, space weather, etc.

In the context of Space Surveillance Awareness (SSA), continuous monitoring is needed to track the position of orbital objects. The purpose of continuous monitoring is to maintain the catalogue of the objects. By doing so, the orbital objects can be tracked to provide information about the orbital safety or as a decision support (e.g. the mitigation scenario) for disastrous incident such as collisions or re-entry of the orbital objects. Optical observation can be done by multiple instrument and techniques. One of the most robust ways is by using a fast response telescope [e.g. ref. 2]. To observe orbital objects optically, there are several methods
such as scanning technique, masking technique, subtraction technique, and other additional technique [2]. The easiest way is the scanning technique, the only one of the four techniques only need a single image to process the information of satellite. Since the angular velocity of the orbital objects is higher than the background stars especially the lower orbit one, the fast-moving object relative to the background stars is observed as the streak of lights, makes it able to be done by using most telescope.

The streaks of light need to be processed to extract the information contained in the image. Many techniques of image processing can be done according to the feature of the objects on the image. It will be a tedious process when so many images are obtained in a night observation. Thus, an automatic processing system is needed. The process in extracting useful information on the image has been proposed by many. Some of them focus on how to do streak detection on astronomical images [3-4] or to detect particular objects such as non-geostationary satellites on survey image [5]. The technique used for detection streaks of light is usually the same with the technique to detect linear future such as Point Spread Function (PSF) analyses and Hough Transform [6].

In this paper, we propose a prototype of automatic satellite streak detection, identification and initial orbit determination pipeline from optical observation. We detect the streak of lights on the FITS Image using Hough Transform, a robust method for detecting lines. This method was used in [2] and [6] in a pipeline for photometry to extract flux of orbital objects on composite image of six consecutive frames observed in trailing mode, based on the optical observation done in Bosscha Observatory Indonesia. Second, we extract the coordinate of the edges streaks and use it to do Initial Orbit Determination (IOD). Parameters of the orbit found from the IOD are used for identifying the streaks by comparing them with TLE data. More details is explained on Section 2. Section 3 explained the result of implementation of the pipeline for blind-survey optical observation data. Finally, the last section discusses the conclusion and the future development notes for improving this pipeline.

2. Methods

We design an automatic pipeline which process fits images from astronomical observation to detect the light streaks, extract their apparent position, identify the object, and estimate the IOD. We aim to have a pipeline which able to run automatically with minimum intervention, compatible with most kind of telescope or imager, as well as easy to modify according to specific needs. This pipeline process plate-solved FITS image with necessary information on its header as the input and then generate detected streak properties, identification and orbit estimation information as the output. The image header should include observation date and time, exposure time, WCS plate-solve information and observatory location.

We implement the pipeline within python environment. We use numpy [7] package for standard mathematical calculation. astropy [8-9] is used for FITS import and astronomical calculation. Image processing and Hough transform are done using OpenCV [10] and scikit-image [11]. TLE information is obtained from space-track1. Apparent satellite position is calculated using skyfield [12]. IOD is calculated using orbdetpy2, a python wrapper for orekit [13].

In principle, any FITS image can be processed in order to detect the streaks. However, it is better to do pre-processing before it processed to give the better result. For example, images should be calibrated instrumentally to obtain a good image with less noise. Even after noise removal, the existence of unwanted noise or error sources is sometimes inevitable. Thus, we conduct median blur before thresholding to eliminate noise (e.g. dark noise or hot pixels) in the images. The images are then converted into binary image by adjusting threshold into the optimum level. The adjustment is done by iterating the threshold value from the "brighter" (e.g. small value) to "darker" (e.g. large value) and choose the optimal value where the background is dark and the faint object is sufficiently bright. This method enables us to keep the appearance

---

1 www.space-track.org
2 ut-astria.github.io/orbdetpy
of faint streaks in binary images.

Before removing remaining noise and stars, we operate morphological transformation to reunite any close object in binary image which separated due to noise. This usually occurred to faint streaks. Then all objects with small area is then removed from image, resulting clean streak (figure 1). Then we operate skeletonization algorithm to retain a line or connectivity feature of the object. The resulting image turn any object into 1 pixel line, so we operate Gaussian blur to the image to turn the 1 pixel line into Gaussian line profile. This enables a better line feature-detection in the next step.

We are using probabilistic Hough-transform from OpenCV to detect line feature and extract the position of both of its ends. In a Hough line transform, set of pixels \((x_i, y_i)\) in an image is transformed into Hough space \((\rho, \theta)\) via,

\[
\rho = x_i \cos \theta + y_i \sin \theta.
\]  

The key of Hough-transform is turning a unique straight line into a single unique point in Hough space, where a unique single point into a unique sinusoidal line in Hough space. A single line is determined by intersection point of sinusoid line in Hough space. The result of detection is \((x_j, y_j)\) and \((x_k, y_k)\) of each end of detected streak. All similar line with very close position (e.g. \(|(x_j, y_j) - (x_k, y_k)| < \text{FWHM}\)) is filtered out into a single line. The pixel position of the line ends \((x_{j,k}, y_{j,k})\) is then transformed into equatorial coordinate based on WCS information from image header by using astropy.

All of the detected streaks is then identified by matching streak position with the calculated position of known satellite from two-line element set (TLE) information using SGP4 model [14] on skyfield. Each streak then labelled by corresponding matched satellite name, where the unidentified streak remains in indexed format. We perform the Initial Orbit Determination (IOD) for each streak by interpolating both time and position linearly. Both of detected streak end must visible inside the frame to obtain correct position and time information. Orbital parameter calculated by using Laplace method [15] via orbdetpy. However, due to a very short observation arc (equal to exposure time), the orbital parameters accuracy will less than what is needed to update TLE. The calculated IOD here may useful for preview purpose, as well as cataloging the unknown object based on its orbital occupation.

Figure 1. Image processing flowchart from original image to Hough line profile detection result.
Figure 2. Orbit determination and satellite identification process. We match the apparent position of the streak with the calculated apparent position from TLE database at mid-exposure. Then we identify the satellite by matching the position with certain radius tolerance.

Table 1. Streak detection and object identification performance result. The binary classification is compared to visual inspection of image data.

| Streak detection       |          |
|------------------------|----------|
| Number of images       | 85       |
| Number of detected streaks | 196   |
| True Positive (TF)     | 123      |
| False Positive (FP)    | 53       |
| False Negative (FN)    | 21       |
| Identical lines        | 117      |
| F1 score               | 0.76     |

| Object identification  |          |
|------------------------|----------|
| Identified streaks     | 93       |
| Unique objects         | 48       |
| Ambiguous detection    | 21       |
| Unidentified           | 5        |

The pipeline automatically scans all fits images within desired folder. The pipeline will result two ASCII files: One containing image properties and the other one containing streak position, timing properties, and orbital parameter properties, as well as identified satellite properties and its calculated state vectors. The pipeline also generates a folder contains original images converted to jpg images with streak detection and TLE satellite position overlay.

3. Result and discussions

We tested the pipeline using optical survey data around celestial equator where number of streaks had been detected visually in images. The instrument is Takahashi FSQ106ED telescope with SBIG STF8300-M camera on 19 September 2020 at National Observatory Operational Office, Kupang regency, Indonesia. The telescope has 106 mm of diameter on Paramount MyT mounting with 1.1 deg field of view. The observation is done by utilizing ACP Observatory robotic telescope control software, which provides all required information on FITS header. We process 85 images along celestial equator in the area between RA 22h-1h. Performance summary can be seen in Table 1.

Based on the result on Table 1, the sensitivity \( \frac{TP}{TP+FN} \) of streak detection is 0.85 with F1
The International Symposium on Space Science 2021 (ISSS 2021)  
Journal of Physics: Conference Series  {2214} (2022) 012018  
IOP Publishing  
doi:10.1088/1742-6596/2214/1/012018

score \( \frac{2TP}{2TP + FP + FN} \) is 0.76. This value not only corresponds to the performance of detection with respect to streak significance, but also to the ability of the pipeline adapt to the various streak shape, image background value, noise, and interfering nearby objects. These aspect should be fine tuned to obtain the optimal detections and accuracy. Several apparent limitations are shown on Figure 3.

Figure 3. Known limitations of the streak detection and identification.

The pipeline compares apparent position of the streak with calculated satellite position with data from space-track. About 47% of detected streak is identified based on 3778 space-track TLE objects at the closest epoch with observation time. All of them is GEO satellites. About half of them is the unique satellites due to multiple frames detection for a single satellite caused by recurrence observation at the same horizontal position in the survey data. The Laplace algorithm would also not able to solve the IOD from certain false detection streak, thus the false positive detection can also be filtered out by considering their IOD convergence. Most of these streaks is identified and consistent compared with TLE data. However, the detection yields an ambiguous result for crowded area where the satellite distance is too tight. The satellite name remains in the indexed format for this case and unidentified case.

The result of orbital determination accuracy is shown on figure 4. The average image exposure time is 20-30 seconds. It means that the observation arc for each IOD is equal to exposure time. The radial distance of the estimated state positions is overestimated 2.7% in average, with standard deviation around 1500 km. The orbital velocity resultant of the estimated state vector is similar to TLE in average, with standard deviation about 130 m/s. This amount of error is likely due to a very short observation arc. However, this state vectors information is still useful for quick look and cataloguing purpose.

4. Conclusions

We have built an automated pipeline for light streak detection, identification and initial orbit determination from telescope images based on python. Due to nature of the flexibility of FITS file, the pipeline will work on any star-tracking astronomical images, as long as all required information are available on the FITS header and both of streak ends is visible inside the frame. This make the pipeline can be utilized on many telescopes including amateur grade telescope.

The pipeline has been tested with 85 FITS images, where 196 light streaks are detected with 0.85 sensitivity. This value can be improved by increasing signal-to-noise ratio of the streak and optimizing parameter values in the thresholding step. About 93 light streaks are automatically identified based on space-track TLE information with 48 unique GEO object. There is still a number of ambiguously identified satellite for crowded area where satellites reside in a close proximity to each other. The result orbital parameter is imprecise for a single image due to a very short observing arc. However, it is useful for preview and cataloguing. The output data is
Figure 4. IOD comparison to TLE data. (a) Radial distance error distribution with respect to TLE and (b) resultant velocity error distribution with respect to TLE. Note that radial distance is overestimated for about 1000 km, where resultant velocity difference is Gaussian around zero with respect to TLE data.

“level 1” data, therefore it needs to be combined with more observation and from different site to improve accuracy and precision via triangulation.

Acknowledgements
The 10-cm Robotic Telescope is operated on the Kupang Regency in the Timau Observatory Operational Office, Indonesia. This research made use of Astropy, a community-developed core Python package for Astronomy [8-9]. We acknowledge the use of the Space-Track database, OpenCV, scikit-image, orbdehtpy, OREKIT, and SkyField packages.

References
[1] Space Environment Statistics - Space Debris User Portal URL https://sdupeкосesa.int/disco-web/statistics/
[2] Lass-Bourez et al. 2009 Advances in Space Research 44 1270–1278 ISSN 02731177 URL https://linkinghub.elsevier.com/retrieve/pii/S0273117709003895
[3] Bengtsson Bernarder K 2014 A method for detecting resident space objects and orbit determination based on star trackers and image analysis
[4] Virtanen et al. 2016 Advances in Space Research 57 1607–1623 ISSN 02731177 URL https://linkinghub.elsevier.com/retrieve/pii/S0273117715006663
[5] Schilcknecht T 2007 The Astronomy and Astrophysics Review 14 41–111
[6] Priyatikanto R et al. 2015 Jurnal Sains Dirgantara 12 ISSN 02731177 ISSN 1432-0746 URL https://www.nature.com/articles/s41586-020-2649-2
[7] The Astropy Collaboration and (Astropy Contributors) 2018 The Astronomical Journal 156 123 ISSN 1538-3881 URL https://iopscience.iop.org/article/10.3847/1538-3881/aabc4f
[8] The Astropy Collaboration 2013 Astronomy & Astrophysics 558 A33 ISSN 0004-6361, 1432-0746 URL http://www.aanda.org/10.1051/0004-6361/201322068
[9] Itseez 2015 Open source computer vision library https://github.com/itseez/opencv
[10] van der Walt et al. 2014 PeerJ 2 e453 ISSN 2167-8359 URL https://peerj.com/articles/453
[11] Rhodes B 2019 Skyfield: High precision research-grade positions for planets and Earth satellites generator (Preprint 1907.024)
[12] Maisonneve L, Pommier V and Parraud P 2010 Orekit: An open source library for operational flight dynamics applications 4th International Conference on Astrodynamics Tools and Techniques pp 3–6
[13] Vallado D and Crawford P 2008 Sgp4 orbit determination AIAA/AAS Astrodynamics Specialist Conference and Exhibit p 6770
[14] Pastor A, Sanjurjo-Rivo M and Escobar D 2021 Advances in Space Research 68 2677–2694