Simon Hatcher, Ashfaq Ahmed, Minki Kim, and Monica Martinez Wilhelmus
Satellite Overpass Identification Tool
20 April 2022

# SOIT: Satellite Overpass Identification Tool

## Motivation and Background:

In polar science, remote sensing techniques enable the quantitative analysis of the ice cover at unprecedented spatial resolution and acquisition periods. Identifying satellite overpass times is essential to process satellite data and compute ice cover metrics with high accuracy.

Traditionally, overpass time retrieval for the Aqua and Terra satellites is accomplished using the Earth Observing System Data and Information System (EOSDIS) of the National Aeronautics and Space Administration (NASA). For each date, retrieving the overpass time involves (1) assessing the area of interest (AOI), (2) estimating which Aqua and Terra orbit paths come closest to the AOI, (3) finding the nearest time indicated on the orbit path, and (4) manually recording this time. This process is prone to human estimation error and is highly tedious.

The process is significantly simplified using the Orbital Prediction and Overpass Tool (OPOT) of the Atmospheric Validation Data Center of the European Space Agency. This tool takes a user-supplied start date, end date, and latitude and longitude of a point of interest (POI) in the AOI and returns pass times for Aqua and Terra satellites in CSV format for each date in the range. Nevertheless, the OPOT has multiple limitations. Users are limited to retrieving pass-time information for 28 days at a time to reduce demand on the site. Due to its web-based nature, these 28-day retrievals are slow, and multiple consecutive 28-day retrievals for an extended date range are prohibitively slow. Furthermore, the website does not determine the closest pass or filter out nighttime passes (where no data is collected); this must be done in a separate program.

The Satellite Overpass Identification Tool program aims to rapidly retrieve the daytime overpass time of the Aqua and Terra satellites closest to a POI for a given range of dates. Using Skyfield, an existing Python orbital mechanics package, and Space-Track.org, an online source of satellite data, the precise overpass times for Aqua and Terra are determined quickly for an extensive range of dates. The algorithm is not only limited to retrieving the data of the Aqua and Terra satellites. It may be easily modified for other instruments (e.g., Aura).

## Algorithm:

Using the Python Requests package, based on the methods of Stokes (2019), a series of two-line elements (TLE) for Aqua and Terra is retrieved from Space-Track.org within a POI in the date range of interest. The TLE fully describes the location and orbit of the satellite at a given time (epoch). Skyfield is used to extrapolate the precise rise, set, and overpass locations and times for each day. The result is a list of overpasses near the POI on the date of interest. Aqua and Terra have ascending and descending passes, respectively, during the local day and descending and ascending passes, respectively, during the local night. This information is used to select the daytime overpass closest to the POI and discard any nighttime overpasses. The process is repeated for all dates.

**Requirements:**

1. Download the latest version of Python (Python3). This program was developed in PyCharm and is publicly available for Mac, Linux, or PC here. Alternatively, you can run the program from the terminal of any machine with a Python 3 installation. You will also need the following packages: requests, json, configparser, numpy, csv, math, Skyfield. You can install them easily by running the following command in the terminal: <pip3 install (package name)>. For example, to install Skyfield: <pip3 install skyfield>.
2. Create an account at Space-Track.org (open-source).
3. Make sure to have a stable internet connection.

**Instructions:**

- First, populate the configuration file SLTrack.ini:
    - Enter your Space-Track.org username and password[1].
    - Specify a start date and an end date (MM/DD/YYYY). Satellite overpass times are retrieved from (and including) the start date up to (and including) the end date.
    - Specify the latitude and longitude of the POI. These must be entered in degree decimal form (e.g. -18.535), not in sexagesimal degrees (degree, minute, second). Note that negative values are used for the West and South directions.
- In a preferred local terminal with the Python installation, navigate to the directory containing the program and the configuration file, and run the program in Python3 by typing: python3 pass_time_v2.py
- After execution, you will retrieve a three-column CSV file from the directory. Each row contains a date and two times corresponding to the overpass times in Coordinated Universal Time (UTC). These are defined as the times when the Aqua and Terra satellites passed closest to the POI on a given date. Only daytime overpasses are provided.

**Example:**
**[Configuration in SLTrack.ini]**
username = xxx@xxx.com
password = xxxxx

start date = 04/01/2021
end date = 04/02/2021

latitude of interest = 72
longitude of interest = -18

**[Output]**

[1]THESE FIELDS ARE NOT SECURE OR PROTECTED IN ANY WAY. THIS PASSWORD SHOULD NOT BE USED FOR ANY OTHER ACCOUNTS .

**Acquired satellite overpass time:** (Aqua) 13:08:03 (Terra) 12:47:55

For comparison, this is the information you would find in EOSDIS:

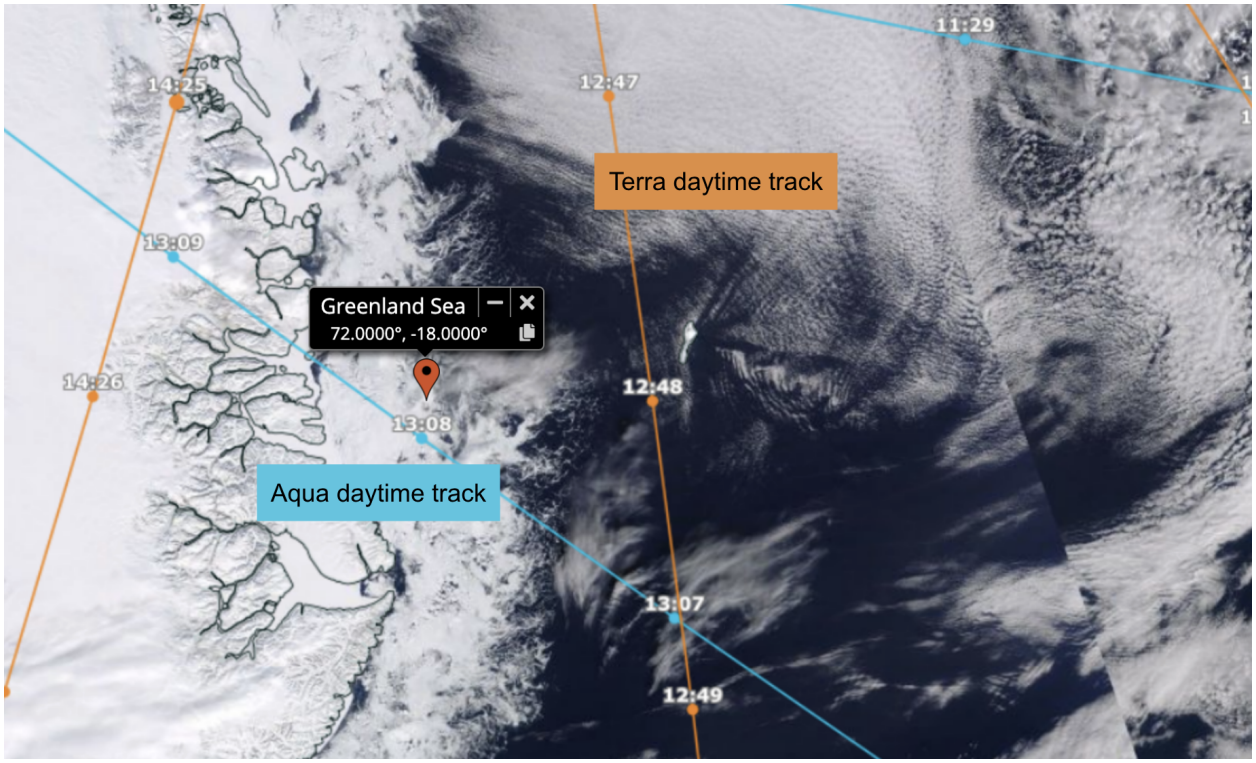


**Figure.** NASA Worldview image near the Greenland Sea (72°N 18°W)

**Acknowledgments:**
This work was funded by the Office of Naval Research (awards N00014-20-1-2753 and N00014-19-1-2421) and would not have been possible without the use of imagery from NASA's Worldview Snapshots application https://wvs.earthdata.nasa.gov/, part of NASA's EOSDIS. We gratefully acknowledge Marjorie Bradley for her insightful comments.

**References:**

Lopez-Acosta, R., Schodlok, M. P., & Wilhelmus, M. M. (2019). Ice Floe Tracker: An algorithm to automatically retrieve Lagrangian trajectories via feature matching from moderate-resolution visual imagery. *Remote Sensing of Environment*, *234*, 111406. doi:10.1016/j.rse.2019.111406

Stokes, A. (2019). *SLTrack.py* [Simple Python app to extract Starlink satellite history data from space-track.org into a spreadsheet]. https://www.orbiter-forum.com/threads/finding-starlink-orbital-histories-via-python.37574/