

# SATCON2 Algorithms Working Group Report

## Authors

Jonathan	McDowell	(Center for Astrophysics   Harvard & Smithsonian); Chair	USA
Rob	Seaman	University of Arizona: Vice-Chair	USA
Cees	Bassa	Astron	NL
David	Galada Enriquez	Calar Alto	ES
Olivier	Hainaut	ESO	DE
Courtney	Hodgson	The University of Texas at Austin	USA
Matt	Holman	CfA	USA
Moriba	Jah	The University of Texas at Austin	USA
JJ	Kavelaars	CADC UBC	CA
Keith	Poletti	Texas	USA
Morgan	Schmitz	Princeton/Vera C. Rubin Observatory	USA
Pat	Seitzer	UMich	USA
Jan	Siminski	ESA	DE
Rachel	Street	LCO	USA
Hossen	Teimoorinia	CADC	CA
Heidi	Thiemann	OpenU	UK
Connie	Walker	NSF's NOIRLab; SATCON2 Co-Chair; Ex-officio WG Member	USA

# Table of Contents

<b>Introduction</b>	<b>2</b>
<b>1. General Software Considerations</b>	<b>4</b>
1.1 Distribution and Documentation	4
<b>2. Test Data Suite</b>	<b>5</b>
2.1 Image test data for TrailMask	5
2.2 Satellite pass test data for PassPredict	6
2.3 Other Test Data	6
<b>3. Software projects</b>	<b>7</b>
3.1 SATCON1 recommendation 1: TrailMask	7
3.1.1 Inputs and outputs	7
3.1.2 Modes	8
3.1.3 Relevant existing software	10
3.1.4 Future algorithms: deep learning	11
3.2 SATCON1 recommendation 2: PassPredict	14
3.2.1 Inputs and Outputs	15
3.2.2 Modes	16
3.2.3 PassPredict: Considerations for the ephemeris database	16
3.2.3.1 Orbital Solutions	17
3.2.3.2 Accuracy, maneuvers, and operator data	18
3.2.4 PassPredict simple mode: detailed requirements for inputs	18
3.2.5 PassPredict: algorithm	19
3.2.6 Available software for PassPredict simple mode	20
3.2.7 PassPredict advanced mode — considerations	20
3.3 SATCON1 recommendation 3: Simulation Tools	21
3.3.1 EphemSimulate	21
3.3.1.1 PassProbability	22
3.3.2 ImageSimulate	24
3.3.3 TrailSimulate	25
3.3.4 TrailAssess	25
3.3.5 Simulation assessment	26
<b>4. Implementation Timescale</b>	<b>27</b>
<b>5. Cross-Group Coordination</b>	<b>27</b>
<b>6. Conclusions</b>	<b>28</b>
<b>References</b>	<b>29</b>
<b>Appendix A: Ephemeris file formats</b>	<b>31</b>
File formats	31
1.1.1 Example CCSDS OEM	31
1.1.2 Example CPF	32
1.1.3 Example NASA Modified ITC Ephemeris format	32
<b>Appendix B: Glossary of Abbreviations</b>	<b>33</b>

# Introduction

This report is part of a collection of Working Group Reports from the [SATCON2](#) Conference.

The published report following SATCON1 (Walker et al., 2020) detailed 10 recommendations, three of which apply to the development of software; these three are the focus of the SATCON2 Algorithm Working Group:

## **Recommendation 1**

*Support development of a software application available to the general astronomy community to identify, model, subtract, and mask satellite trails in images on the basis of user-supplied parameters.*

## **Recommendation 2**

*Support development of a software application for observation planning available to the general astronomy community that predicts the time and projection of satellite transits through an image, given celestial position, time of night, exposure length, and field of view, based on the public database of ephemerides. Current simulation work provides a strong basis for the development of such an application.*

## **Recommendation 3**

*Support selected detailed simulations of the effects on data analysis systematics and data reduction signal-to-noise impacts of masked trails on scientific programs affected by satellite constellations. Aggregation of results should identify any lower thresholds for the brightness or rate of occurrence of satellite trails that would significantly reduce their negative impact on the observations.*

We have attempted to transform these SATCON1 recommendations into a specific set of high-level software requirements with provisional names for convenience of reference. We note that each of these is a fairly major software effort if they are to be robust enough to support the community. However, in some cases relevant software already exists, and this document identifies those packages.

In brief, the SATCON1 recommendations call for the ability to flag, mask and repair satellite trails affecting astronomical data (a software tool we call TrailMask), to predict when satellite trails may or will affect specific observations (which we call PassPredict), and to simulate the effects of satellite trails so that the community can assess the scientific impact of those effects on astronomical research.

Our main focus is on ground-based optical images of all kinds. However, we also considered space-based images and spectroscopy. We did not consider the (important) effects of satellite constellations on radio astronomy, although the PassPredict tools should work for single-dish radio observations to the extent that sidelobes are not important.

We are mostly concerned with the large low-Earth orbit (LEO) satellite constellations. However, spacecraft at near-lunar distance are regularly seen by asteroid surveys, so we should consider MEO and GEO (medium Earth orbit and geosynchronous Earth orbit) cases too. Note that the fainter magnitude of high-orbit satellites is offset by their lower apparent angular velocity, leading to larger effective exposure time on a streak pixel.

In Figure 1 we note that the counts in a satellite streak will tend to be independent of exposure time, and so the measured magnitude of the streak will be fainter for longer exposures.

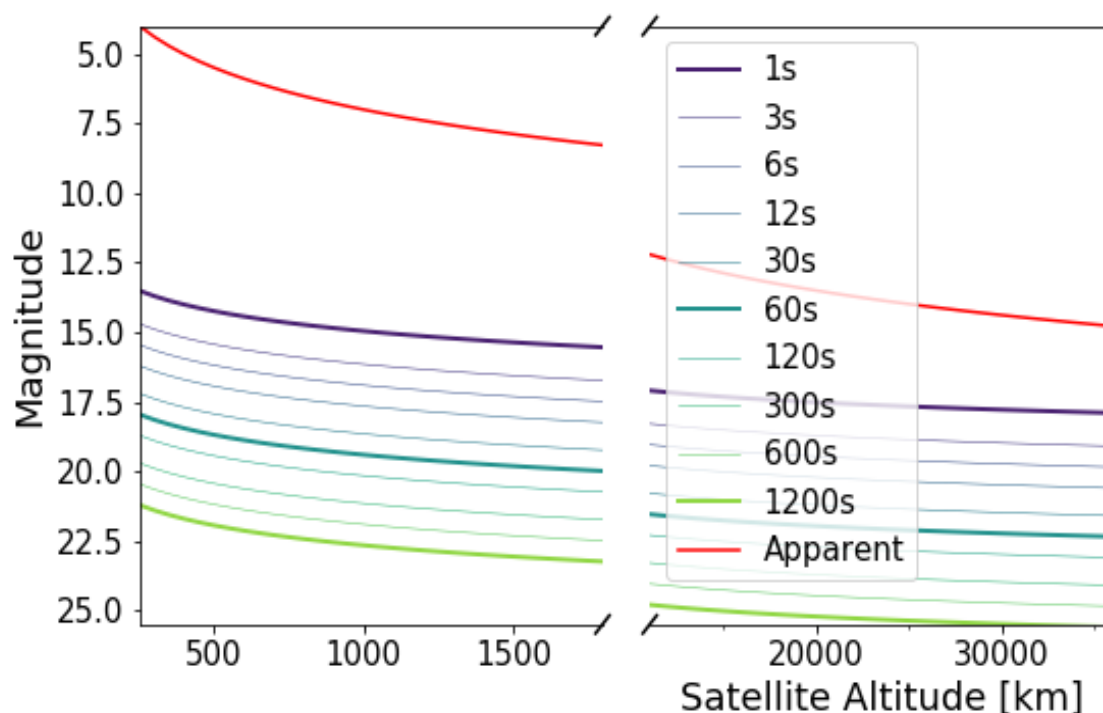


Figure 1. Effect of trailing on the effective magnitude of a satellite. Red: visual magnitude at zenith of an example satellite as a function of orbit altitude. Blue to Green: observed magnitude of the same satellite accounting for trailing, for a series of increasing exposure times and assuming a 1-arcsecond resolution element. In a given telescope/instrument, as exposure time increases, the number of counts detected from a faint (e.g., 15th magnitude) star will increase, but the number of counts in the satellite trail will not, assuming that the satellite crosses the field of view in a time that is short compared to the exposure time. Thus, the apparent brightness of the satellite trail will be comparable to stars of increasingly faint magnitude with increasing exposure. We assume this is a small telescope, so the spatial extent of the satellite (defocus + resolved size) is not accounted for.

The effects of streaks on optical imaging data were discussed in the SATCON1 Report (Walker et al., 2020). In this working group we also discussed the effects on other kinds of data, especially spectroscopy. Low-spectral-resolution fiber spectroscopy is especially vulnerable — the effect of a satellite streak is to add a solar spectrum to the target spectrum, and in the absence of any spatial information it may be hard to spot that your data has been affected. The limiting magnitude of low to medium resolution spectrographs are typically in the 20-25 range, comparable to the effective magnitudes of many satellites. ESO is planning a system with 3000 fibers at low resolution, and they estimate that the satellite contamination (when it occurs) will be up to 5-10 sigma above noise. This could be bad; 1000 sigma is easy to spot, 0.1 sigma can be ignored, but the intermediate range is difficult to notice and yet affects the scientific result. Higher resolution spectrographs have much shallower limiting magnitudes (15-20, even on large telescopes), and will therefore be essentially immune to the contamination by all but the brightest satellites. Finally, although exoplanet transit spectroscopy already has the problem of subtraction at the limits of S/N, its high spectral resolution will prevent significant contamination problems.

# 1. General Software Considerations

The astronomy software community is investing heavily in Python and in particular in the astropy suite. While we do not exclude other languages, applications developed in Python and compatible with astropy are more likely to be easily installed and usable by a broad audience.

External dependencies are sometimes necessary but each extra one adds maintenance overhead and often limits the potential user base; they should therefore be used judiciously.

Software-savvy astronomers will want to access the software by calling libraries (typically Python ones). However, less software-aware astronomers (both professional and amateur) will need command-line or web-based end-to-end tools which wrap these libraries in a simple interface. We must support both of these communities. In particular, a simple browser-based interface to the PassPredict and TrailMask tools discussed below (at least in their simple mode) is strongly recommended. We should also provide interfaces to planetarium applications (World Wide Telescope, OpenSpace, Stellarium).

If the software is to be used widely by astronomers, it should if at all possible be open-source, free, and free of restrictive licences. We should support a software ecosystem in which centrally developed reference implementations may exist, but interfaces are simple and well documented so that alternative implementations can be swapped in — this will allow us to leverage innovation by the community.

We encourage support of International Virtual Observatory Alliance (IVOA) protocols, and specifically pyVO, for retrieval of test datasets (and possibly of satellite prediction data if appropriate protocols exist). However, programs should always also allow import of datasets from a local disk.

Where appropriate (e.g., for satellite reflectance models) the software architecture should allow for user-model plugins (i.e., users can write their own model and have the software use that instead of the presupplied one).

## 1.1 Distribution and Documentation

The software will require user documentation and support. The obvious place to serve as a portal for software and documentation is the SatHub proposed in the Observations Working Group Report. We also recommend the development of related educational materials such as lesson plans to engage the school and university student communities.

## 2. Test Data Suite

The working group concluded that early development of a test dataset repository is a priority. Standard test datasets covering a range of cases will be needed during software development to validate algorithms and to compare the performance of different algorithms. Test datasets may also be of use to the Observations Working Group. The suite should be as small as possible (to be manageable) while still covering the needed range of test cases.

### 2.1 Image test data for TrailMask

The image test suite will be used to test TrailMask. It should include actual examples of images with satellite streaks, including the following cases:

- Large and small fields of view
- Large and small angular pixel sizes
- Short and long exposures
- Low and high background
- Professional and amateur telescopes
- Bright and faint streaks
- LEO and MEO/GEO satellite streaks
- Crowded and sparse fields
- Streaks that cross each other
- Optical and infrared data
- Mosaic frame and IFU (integral field unit) datasets
- Polarimetry data
- Simulated datasets (or real datasets with simulated streaks added) as needed — Vera C. Rubin Observatory has such data that could be used

Test cases should also include fiber spectroscopy and polarimetry examples. They should also include examples of false positives (e.g., comets/asteroids). The test case images should ideally be bias-/dark-subtracted and flat-fielded, although we should also include some uncalibrated images as test cases. It would be best if every test case consisted of a pair of images — one with the trail and one without, to test how well TrailMask works. All test cases should each be accompanied by a text description indicating their relevance (use cases etc.).

The test cases should also cover a range of telescopes and instrument types, including:

- All sky camera with a field of view > 150 degrees
- Commercial astrophotography lenses paired with DSLR, CCD, CMOS detectors
- Large CCD: 2k x 2k or larger with telescopes of various apertures from 1 to 30 metres (singly or in mosaics).

Image cases should include all metadata needed to perform different kinds of analysis — in particular observation time and pointing direction to support streak identification use cases. Use of at least the IVOA ObsCore DataModel is recommended as this will ensure that the developed software will work within a broad ecosystem. The metadata list should specifically include:

- Site (longitude, latitude, altitude in WGS84)
- Full image World Coordinate System (WCS) (ICRS etc; includes pixel scale.)
- Exposure start time and duration
- Filter (with documentation link to transmission curve)
- Optical setup (focal length, aperture, type)
- Photometric zeropoint (pixel values to Jansky or magnitude?)
- Gain noise / read noise

## 2.2 Satellite pass test data for PassPredict

A satellite pass test suite is needed to test PassPredict. The ObsCore data model should also be used here. Each test case should include:

- The observing conditions (telescope and camera parameters, pointing direction, date and time)
- A fixed test satellite database from which satellite ephemerides may be extracted, or (for some use cases) a single satellite ephemeris prediction (when you don't want to test the database part)

We should be able to (retro-)predict the passes of a few specific satellites at a specific epoch over a specific observatory and (if possible) predict their brightness.

## 2.3 Other Test Data

Test cases of secondary priority may include:

- Fiber spectroscopy and slit spectroscopy cases;
- Radio astronomy cases;
- Infrared astronomy cases; or
- TrailMask user-supplied cases, particularly if case is interesting/pathological.

We have not yet discussed what test cases would be needed for the simulation tools in the third SATCON1 recommendation above.

**Effort required:** establishing a repository where TestData can be organized and discovered should be done at one of the established astronomy data repositories. An upload and metadata registration system will require about 2 FTE (split across different expertise groups) with ongoing support requiring 0.25 FTE

## 3. Software projects

Here we discuss one by one the individual software algorithms that we see as responding to the SATCON1 recommendations. It is likely beyond the scope of this working group to choose or down-select a particular approach but we can provide some guidance on how such a selection might be made.

### 3.1 SATCON1 recommendation 1: TrailMask

*Support development of a software application available to the general astronomy community to identify, model, subtract and mask satellite trails in images on the basis of user-supplied parameters.*

As noted in Section 1, both programmatic and web-based interfaces should be provided. The latter will be of particular use to the hobbyist community.

#### 3.1.1 Inputs and outputs

Required inputs:

- a) Image(s) where trails should be identified
- b) Image parameters (field of view, pixel size, flux calibration; would usually be in the image's header)
- c) Trail search parameters (width ranges, signal-to-noise ratio etc.; should come with reasonable default values if derivable from input b)

Additional, optional inputs (depending on the mode TrailMask is run in):

- d) Time and pointing of observation (for seeded mode)
- e) Prior information on where trails are expected to be present (for seeded mode)
- f) Simulated satellite traces planted on real images and also the images without the traces as the training set for deep learning models
- g) Real satellite traces in images — coming from the test data suite and elsewhere.
- h) Images from the same region as g) without the traces as an alternative training set for deep learning models

Outputs (depending on the mode, any combination of):

- 1) Catalog of identified trails, including some parameters (trail brightness estimates and brightness uncertainty estimates, start and end positions, width of the trail, and other parameters to be determined)
- 2) Mask file with flag set for each affected pixel
- 3) Images with trail affected pixels modified to minimize impact on data.



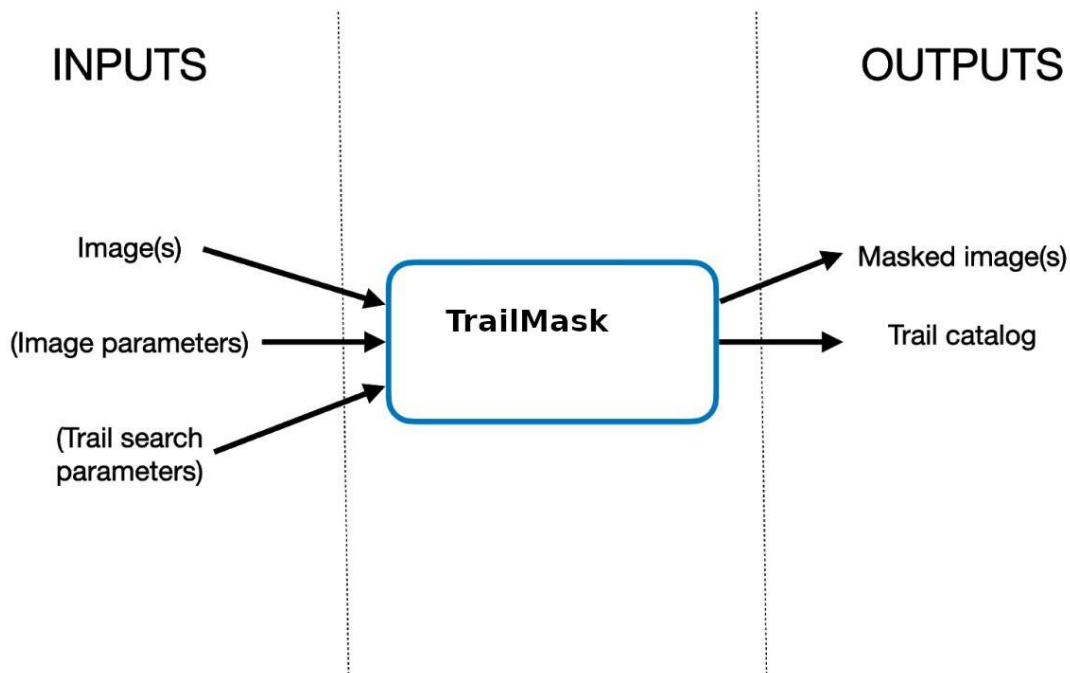


Figure 2. Schematic of TrailMask, when running in its simple configuration. In this example, since the user does not intend to use trail-subtracted images, the only outputs are the identified trail catalog and masked images.

### 3.1.2 Modes

TrailMask should support several modes, using different prior information and different desired outputs.

#### *Prior information*

- Run “seeded” with manual info about trail location and parameters, (optional input e)
- Run “seeded” using the output of PassPredict or similar
- Run “blind” without prior info on where the trails are

These different options ensure that one can deal with trails that were not predicted or were mispredicted.

#### *Desired output*

TrailMask should have the capability to find trails, mask every affected pixel, model the trails, and even minimize the noise signal. However, depending on use case, some or several of these options might be superfluous, or even undesirable. For instance, in many science cases where the noise properties must be very well understood, it is preferable to flag and ignore affected pixels rather than trying to recover them. In those cases, it would of course be wasteful for TrailMask to perform all the computations required to obtain all outputs. It is therefore crucial that the user be able to decide which combination of the various available outputs (1–3) they desire.

#### *Approach*

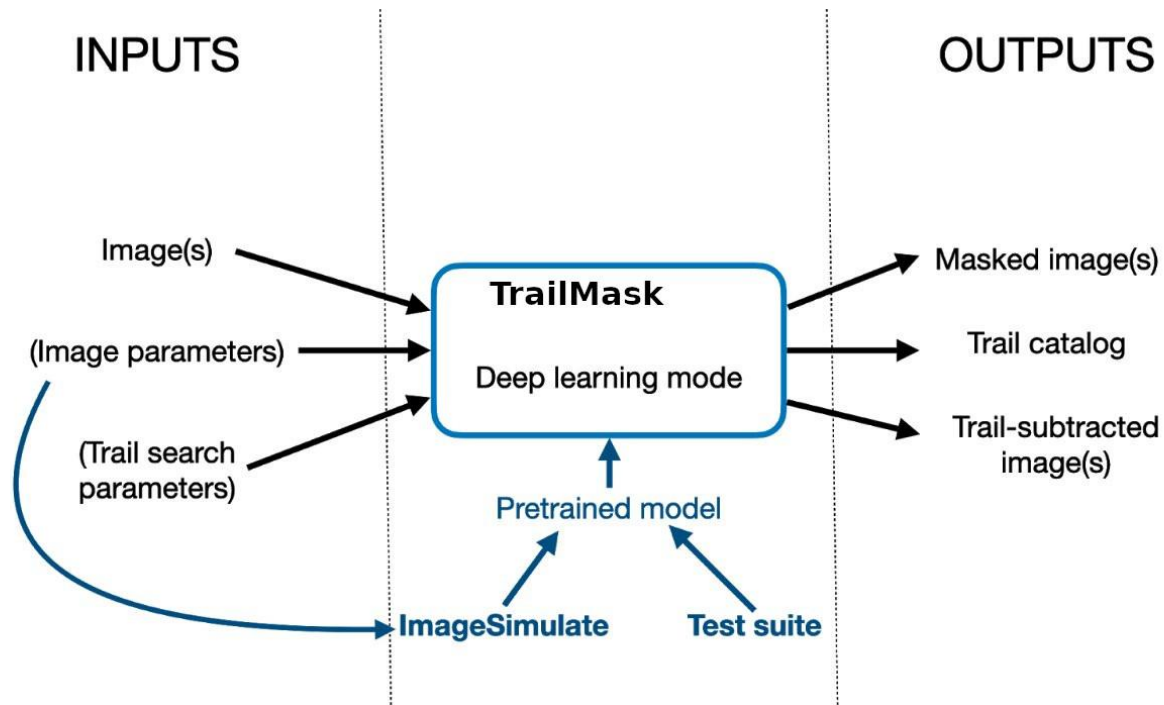


Figure 3. Schematic of TrailMask, when running in deep learning mode. If input images are too different from data used to train the pretrained model, but the user does not provide their own training data, TrailMask can rely on ImageSimulate (section 3.3.2) to provide extra training data.

When several frames of the same part of the sky are available, image differencing can be used to identify trail locations, and a simple median-stacking can be sufficient to remove the trail. TrailMask must also handle the case where only a single image is given as input. A simple, algorithmic approach should be available, and be able to produce satisfactory results for outputs 1 and 2. This could be based on the Hough Transform. However, modified methods may be needed to handle curved trails, which are especially likely to occur in space-based observations. Lastly, a more advanced, deep-learning-based method can be used, allowing for output 3 to be produced (and likely improving the quality of outputs 1 and 2). See Fig. 2 and Fig. 3 for schematic descriptions of these approaches.

**Other considerations.** How will the cutoff transverse to the trail be set? How will the algorithms behave on curved trails? How will ghosting be handled? For non saturated trails, can we assess whether faint sources can be detected under the trail?

A separate program under the TrailMask area might be a spectroscopy analysis tool to detect spectra showing contamination by a satellite spectrum (which will be close in shape to the solar spectrum).

#### Training data

The deep learning approach itself can run in several different modes. A pretrained model will be available as part of TrailMask. If the given input data are close enough for the model to be expected to perform well, this mode can be used without requiring any additional inputs. If not, appropriate training data can be used to generalize the model. These can be provided by the user (optional inputs f, or g+h), found in the test data suite if it contains images from a similar instrument, or be simulated in place by feeding input b to ImageSimulate (section 3.3.2 below).

### *TrailMask and the test data suite*

To keep improving the pretrained model that is shipped with each TrailMask version, data from the test suite can be used. Conversely, in those cases where a user provides their own training data and at their discretion, these could be added to the test suite. This would require a certain level of interaction between TrailMask and the test suite, such that the former can query the suite, and potentially upload new data to it.

The use of processes that rely on databases of satellites will be inherently limited by the lack of availability of accurate orbits. Mitigation strategies for imaging should not rely on weakly available information. In addition, the goal of ‘TrailMask’ is twofold:

- Flag noise from satellite trails to reduce corruption in science analysis.
- Reduce/remove the noise that satellite trails insert into images.

Making images “look pretty” with various blending techniques is less important to the science mission. However, visual inspection of images is often important for analysis. Removal of satellite trails to allow deeper visual inspection (while ensuring the pixels impacted by the removal are flagged) is an important capability.

## 3.1.3 Relevant existing software

- Rubin Observatory’s satellite trail finder maskStreaks.py<sup>1</sup> — all open software.
  - Inputs: images with detection footprints
  - Outputs: images, with an additional mask bit for pixels that fall within streaks
  - Rough outline of the algorithm:
    - Uses Canny filter to make binary image of edges (the user could also provide a binary image instead of an image with detections and bypass this step)
    - Uses Kernel Hough Transform to find clusters of points and fit lines to each cluster
    - Takes sets of nearby lines as identifying the same streak
    - Fits a translated Moffat profile to the final lines
    - Adds mask bits
  - For now, the Rubin Observatory team are using it on the difference between single exposures (point spread function matched warps) and a static sky model
  - The current implementation requires the full Rubin Observatory artillery (the Science Pipelines) to be run, so it is clearly not an off-the-shelf solution. “Derubinizing” the algorithm itself should be a manageable task, if the community decides it is desirable.
- The CADC (Canadian Astronomy Data Center) Image Quality assessment process. In Teimoorinia et al. (2021) , it was presented as a process for the detection of trailed images but the satellite problem is similar.
- MaxiMask<sup>2</sup> is a CNN-based (convolutional neural network) trail identifier
- [Desai et al. \(2016\)](#) propose an algorithm that uses a deep co-added image of the same area of the sky as the exposure of interest
  - This may be too specific to sky survey-type observations to be of general use.
- [Gruen et al. \(2014\)](#) have a publicly available, modified version of SWarp to remove artefacts, including satellite trails.
  - This algorithm also supposes numerous exposures of the same area are available.

---

<sup>1</sup> [https://github.com/lst/pipe\\_tasks/blob/master/python/lst/pipe/tasks/maskStreaks.py](https://github.com/lst/pipe_tasks/blob/master/python/lst/pipe/tasks/maskStreaks.py)

<sup>2</sup> <https://github.com/mpaillassa/MaxiMask>

- StreakDet<sup>3</sup>, a European Space Agency (ESA) software package. It was developed to find space debris streaks, e.g., for on-board processing on an optical payload. It is available under a weak-copy left license and is not open source.
- Cosmic-CoNN (Xu et al., 2021) is a CNN architecture for cosmic ray detection, though as they say it should be easily generalizable to satellite trails. Especially relevant is their proof of generalization to other instruments with minimum input data for retraining once the pretrained model has been trained on a large volume of data (from Las Cumbres Observatory in their case). As they say, *“By expanding our dataset with more instruments from other facilities, we are confident to see an universal cosmic ray detection model that achieves better performance on unseen ground- based instruments without further training.”*

**Effort required.** The effort needed to produce a TrailMask process will be dependent on the path selected. Adoption of the Legacy Survey of Space and Time (LSST) pipeline based trailmask or similar pipeline to a generic environment will likely require around 2 FTE. Simple codes that remove trails via image stacking require nearly no effort but are only effective for stacks.

### 3.1.4 Future algorithms: deep learning

Deep learning/AI methods for both detection and removal of satellite trails are being developed and may provide a highly effective approach to solving the problem of detection and removal of trails.

A deep learning/AI implementation would have the following user modes:

- 1) Pretrained nets
- 2) User supplied training set (one could call the simulation tools discussed below to generate simulated data from observation parameters)

A deep learning generative model has been used by the CADC team to remove moving object trails; it uses the open source tensorflow library. However, designing the model is not easy. A deep learning model can be trained, with relevant data sets, to detect and model various objects in images, such as satellite traces in astronomical images. The models can then be used to remove the trails from the images using deep learning techniques. Tensorflow, an open-source machine learning platform, can provide a foundation for training deep learning models. Keras, a deep learning API (applications programming interface) written in Python, runs on top of TensorFlow's machine learning platform, focusing on enabling fast experimentation and easy implementation. With Keras, a trained model can be used easily as Python code, standalone, or a part of a pipeline.

The deep models that are trained on a single instrument are likely not generically applicable but they can be used as pre-trained models for trail detection on new instruments. Training for the new instrument can then be achieved using a smaller training data set. Two items needed to enable this transfer in learning are a database with some uniformity in accessibility and metadata associated with the training data.

Deep learning methods are also highly effective at learning in lower-dimensional representations, known as latent space representation. Images with similar characteristics lie near each other in latent space. The vector length is considerably smaller than the input image size, providing a compressed representation of the original image by removing complex dimensionality associated with astrophysically uninformative parts of the image space. The latent space vector can also be mapped back into the original space, restoring the original image (i.e., preserving fluxes, with possible random losses). The deep learning

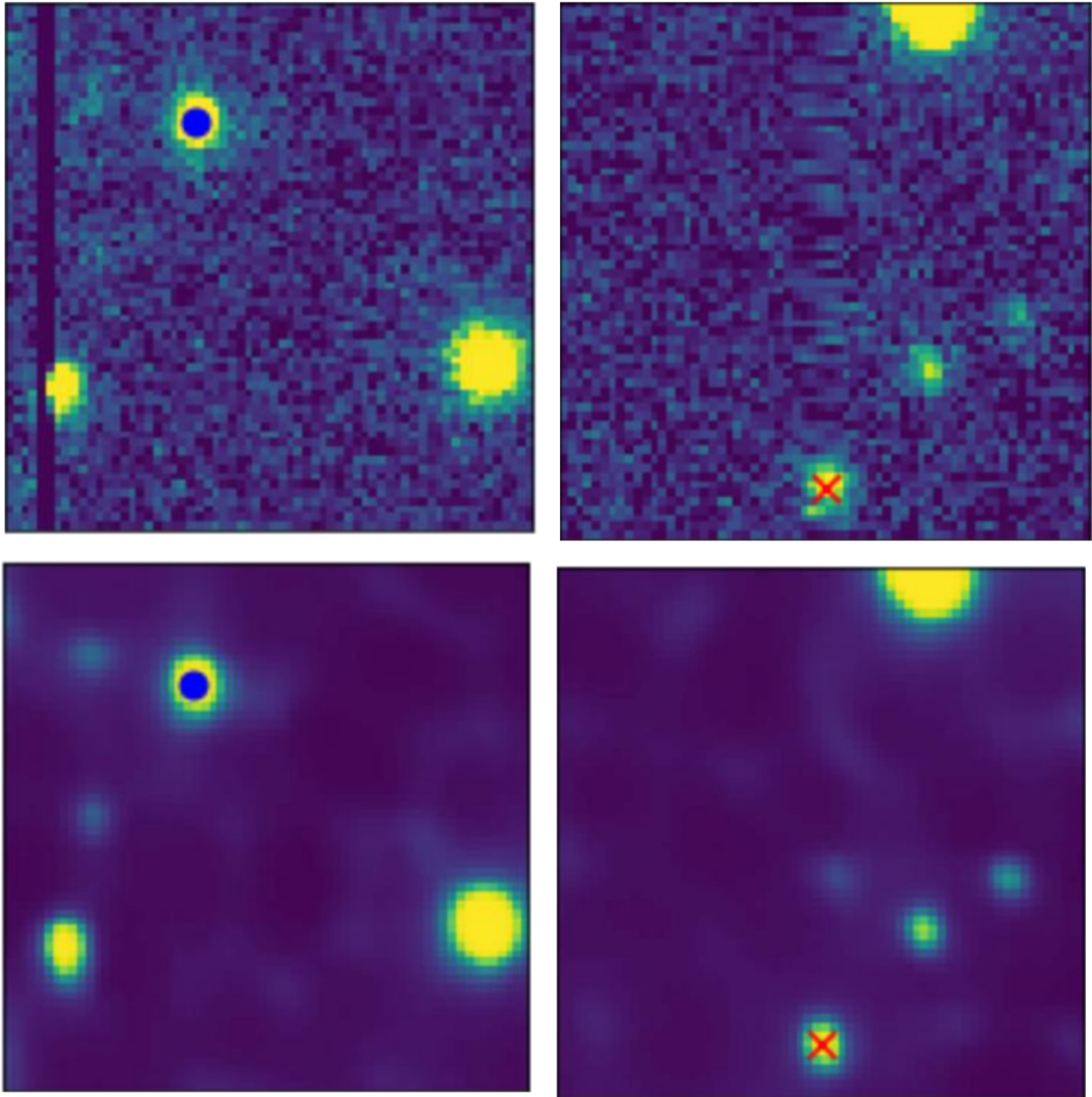
---

<sup>3</sup> <https://conference.sdo.esoc.esa.int/proceedings/sdc7/paper/1018/SDC7-paper1018.pdf>

model that creates the latent representation is trained such that the compressed latent representation contains astrophysically meaningful information. Simultaneously, the latent vector outputs of deep models provide a homogenous input for deep downstream models trained to solve astrophysical problems. A set of deep probabilistic deep models can learn "what is in the images".

Beyond use as homogenous inputs for deep learning, the latent space representations of the images themselves are also highly useful. A remarkable application of using latent space in deep models is to perform arithmetic methods on the latent space data. The algebraic manipulation has a visible manifestation when latent data are decoded back into the original image domain. For example, suppose we have a set of latent space vectors of particular sky images containing contamination from satellite tracks and another set of images for the same sky but without these tracks. The latent space vectors can be subtracted from each other. Once subtracted, the leftover latent data will only represent the satellite tracks. These new latent satellite vectors can act as versatile models; for example, they can be subtracted from other contaminated images to remove undesirable tracks.

As an example of deep learning methods (that can also be generalized to the satellite problems), the top panel of Figure 4 shows two different images with different pixel problems. The blue and red marks show two moving objects. The bottom panel shows the same images where the images have been denoised, and the problems have also been removed without damaging the sources in the images.



*Figure 4. Examples of deep learning methods. In the top row are two different images with different pixel problems. The blue and red marks show two moving objects. The bottom row shows the same images after they have been denoised, and the problems have also been removed without damaging the sources in the images.*

As another example, the deep models can also be used as a content-based recommendation system capable of filtering images based on the desired content (see Teimoorinia et al., 2021). In Figure 5 (a Self Organizing Map), different sources of a set of astronomical images are modeled by a deep model. The model is capable of recognizing images with bad pixels (e.g., the sources similar to node (1, 12)), images with a bad focus problem (e.g., node (2, 1)) or images with satellite tracks (e.g., node (23, 1)).



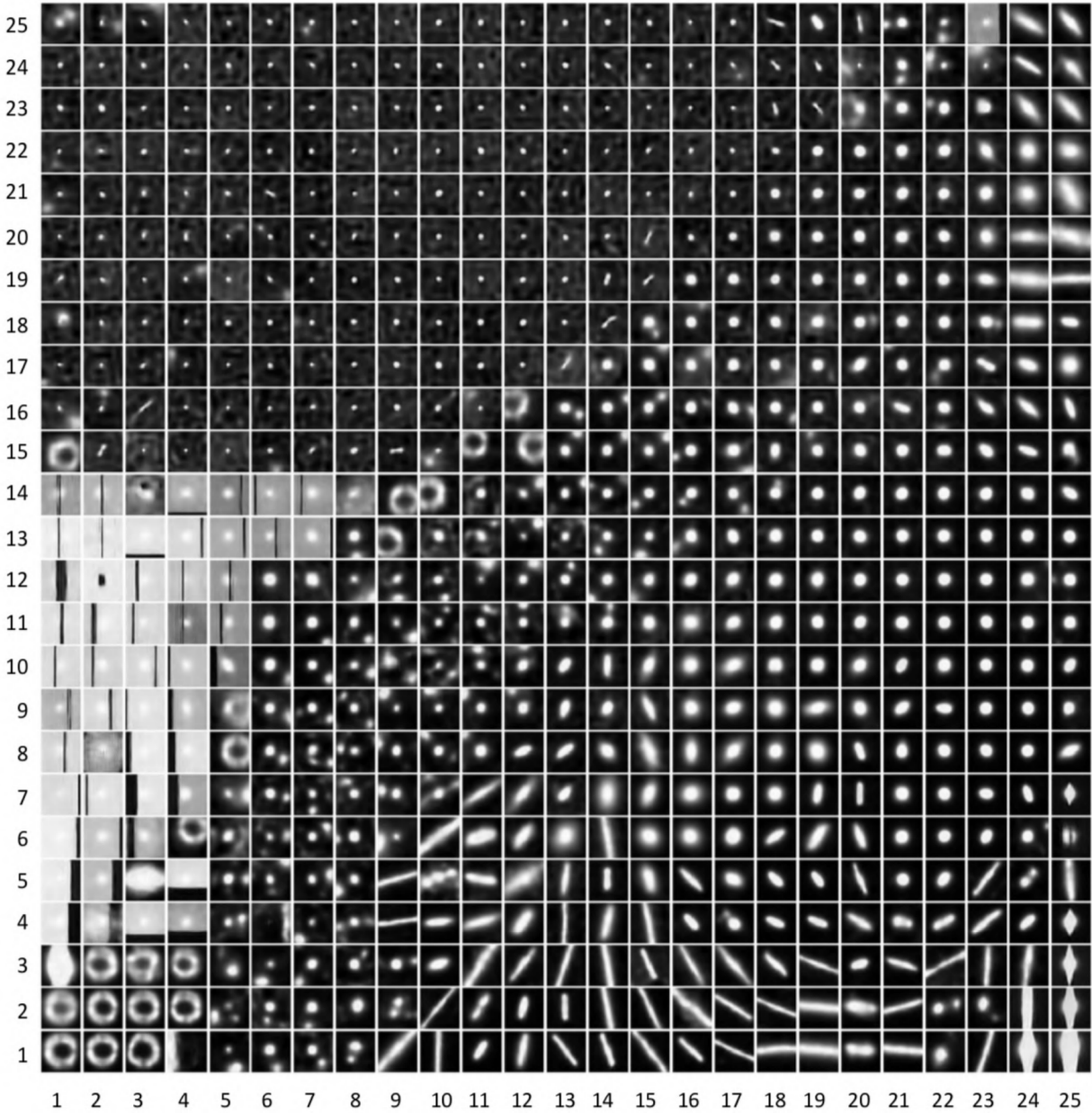


Figure 5. A Self Organizing Map showing different sources for a set of astronomical images modeled by a deep learning model. The model is capable of recognizing images with bad pixels (e.g., the sources similar to node (1, 12)), images with a bad focus problem (e.g., node (2, 1)) or images with satellite tracks (e.g., node (23, 1)).

## 3.2 SATCON1 recommendation 2: PassPredict

*Support development of a software application for observation planning available to the general astronomy community that predicts the time and projection of satellite transits through an image, given celestial position, time of night, exposure length, and field of view, based on the public database of ephemerides. Current simulation work provides a*

*strong basis for the development of such an application.*

A browser-based interface will enable astronomers to predict what satellites will intersect with any given single observation. However, to be effective as a mitigation strategy, astronomers will need to use an interface optimized for planning observations in advance, adjusting either the pointing and/or the timing of the observation to minimize the number of satellite tracks or even (if possible) shifting the observation to a different observatory location/instrument field of view. This will mean multiple queries to the database with adjusted parameters for each observation. Together with the need for intensive observing programs and rapid response to transient events, this implies the need for an interface accessible from programs (i.e., an API) which would handle large batch requests with inputs and outputs in a parseable format (e.g., JSON). Deployment of a queryable system via the IVOA TAP protocol would leverage several existing application tools, such as pyVO. This program will be computationally intensive and may benefit from optimisations (such as grouping calculations of satellites in similar orbits) and from use of parallelization and GPUs.

### 3.2.1 Inputs and Outputs

Inputs:

- A. Ephemeris database (real or simulated) and in-orbit satellite list
- B. Observatory parameters: latitude, longitude, height
- C. Observation schedule parameters (right ascension, declination, date/time, exposure time, field of view, aperture)
- D. Satellite physical and optical properties (bidirectional reflectance distribution function [BRDF] etc.) including software model to predict brightness (advanced mode only)
- E. Streak minimum brightness threshold (advanced mode only)
- F. Photometric bands in which to estimate brightness (advanced mode only)
- G. Possibly, Satellite attitude ephemeris (needed to use the BRDF)

Outputs:

1. Transit list: Satellite catalog number, time, probability, trail parameters+uncertainties.
  - a. Trail parameters include satellite magnitude, trail surface brightness, trail width, trail start and end location.

The predicted trail location on the image may be too uncertain to be useful for actual predictions, but will be required for simulation exercises.

Accuracy requirements on inputs:

- We consider two levels of accuracy: COARSE — enough to say if the field of view is affected; and FINE — enough to say where the streak will be in the field of view (ideally to the pixel level).
- For COARSE accuracy we require about 10-arcminute fidelity, corresponding to about 1 km at the satellite. For FINE accuracy we would like arcsecond fidelity, corresponding to meter-level knowledge of the cross-track satellite position.
- The along-track angular velocity of the satellite is large, the time to cross the field of view being on the order of a second. The along-track requirements on the prediction accuracy may be somewhat weaker, but confirming this requires a more accurate calculation involving the transverse angular velocity of the streak in the detector frame.
- Requirements on the observatory parameters are similar to those on the satellite: kilometer for coarse, meter for fine.



- Requirements on the observation parameters include at least 1 second absolute accuracy on the exposure start and end time, to match the coarse requirement for satellite position.

Accuracy requirement on methods (and so outputs):

- For FINE accuracy refraction and aberration must be accounted for.
- For brightness calculations in advanced mode, 0.1 magnitude is probably more than good enough to assess the scientific impact of streaks.

### 3.2.2 Modes

PassPredict will have three modes:

- Simple mode, predicting position but not brightness. This should include support for a browser-based interface as well as an API.
- Advanced mode, predicting brightness of the satellite as well.
- A posteriori mode, for identifying streaks or (importantly) possibly compromised fiber spectra — in archives.
  - Note that *post facto* operator ephemerides are often more accurate than predictive ones, in contrast to two-line elements (TLEs) which are never improved retrospectively.

One could separate out the “position [and optionally brightness] of satellite vs time” part from the “what does the streak look like on the image?” part and make them two separate programs.

### 3.2.3 PassPredict: Considerations for the ephemeris database

The ephemeris database will be an interface to a variety of data inputs:

- The list of satellites to be considered. Each satellite (with a few exceptions that we aren’t interested in) may be labelled by its number in the US satellite catalog. This is a 9-digit integer (currently all but a few objects use only 5 digits, but that’s about to change). Tables are available to look up the satellite name and owner as a function of catalog number, and to look up which catalog numbers are currently in orbit rather than re-entered. Note that for efficiency our database should avoid requesting fresh orbital data for a satellite which has now re-entered, so it needs to keep track of such events.
- The orbital solutions for each satellite.
- A curated list of active satellites and their current status (in orbit, orbit raising, operational, failed, etc.) is desirable. Such lists are currently maintained by unfunded enthusiasts but there is no sustainable project to support this in the long term. One can find the list of all satellites in orbit by filtering the space-track catalog; at a minimum we should provide software to do this and cache it on a weekly basis, rather than running a space-track query every time PassPredict is run. Additional information on the status of each satellite will be needed for assessments of the overall impact of the constellations and such a list should be supported and maintained by the astronomical community. In addition to the usual catalog-numbered satellites, some numbers are reserved for so-called ‘analyst satellites’. These are objects not identified with a specific launch and can be thought of as analogous to unnumbered minor planets. The analyst numbers are arbitrarily re-used. The megaconstellation satellites we are mostly concerned with will probably not be in the analyst list, so it is not urgent to consider them.

### 3.2.3.1 Orbital Solutions

The position of a satellite at a future time is predicted by an algorithm called a “propagator”, using an orbital solution at some given past epoch. For accurate predictions the epoch used should be only a few days old, at most a week, especially in low orbits (since drag effects and atmospheric density are not predictable on long timescales). The propagator that you need depends on the model used to generate the orbital solution. There are two main sources of orbital solutions:

- Operator orbit solutions for their own active satellites determined by active tracking of the satellite radio signal or derived from onboard GPS receivers
- Passive (radar or optical) tracking of satellites, including inactive satellites and debris. This is systematically done by US Space Force 18th Space Control Sqn (18SPCS) and the Russian Space Forces’ SKKP, and is now also being done by commercial companies LeoLabs and ExoAnalytics, and to some extent by ESA. For brighter satellites optical and radio-transmission tracking is also done by hobbyists.

The orbital solutions used are typically one of several types:

- GP (General Perturbations) mean elements. These are time-averaged Keplerian elements using a model called SGP4 which takes a simple drag model and some other perturbations into account. GP elements used to be provided in TLE format but are now also available in JSON and other formats.
- SP (Special Perturbation) state vectors — the state vector (position plus velocity) at an epoch in a particular frame. These state vectors are not directly observed but are derived from orbit fits using high-fidelity force models.
- SP ephemerides — sets of predicted state vectors vs time which can be interpolated using Lagrange or Hermite polynomials.
- Other forms of state vector ephemerides, including those from NASA or the International Laser Ranging Service (ILRS).

The elements or state vectors are given in one of several reference frames. The most common ones are:

- EME2000 (The astrodynamical name for J2000)
- TEME (True Equator Mean Equinox, sort-of-but-not-quite equator of date). Space-Track TLEs are in TEME.
- ITRF (International Terrestrial Reference Frame, rotating with Earth)

Availability of data, all of which is updated approximately daily:

- GP/TLE data are available from 18SPCS via the public website<sup>4</sup> for all satellites except secret US satellites, whose GP/TLE data are instead made available by hobbyists<sup>5</sup>.
- The SP data from 18SPCS are available to operators by special arrangement with Space Force but not to the public. Such a special arrangement usually comes with restrictions that would be incompatible with our needs.
- Some researchers (e.g., M. Jah, UT Austin) have access to LEOlabs data for academic research, but they are in general not free.

---

<sup>4</sup> <https://space-track.org>

<sup>5</sup> notably <https://www.prismnet.com/~mmcants/tles>

- Some operators (notably Starlink, OneWeb, GPS) make their GP/TLE data available publicly. TLE versions of these data are available on T.S. Kelso's Celestrak site<sup>6</sup>.
- SP ephemeris files (several GByte per day) for SpaceX Starlink satellites are publicly available<sup>7</sup>.

One of the conclusions from all our previous and current work is that our proposed PassPredict tools will need orbital information allowing for position accuracy of the order of an arcminute for imaging applications and general purpose evaluation. This is equivalent to knowing the position of a LEO satellite with a precision of  $\sim 200$  m. For spectroscopy, the requirements would be even more stringent, on the order of an arcsecond or  $\sim 5$  m. These levels of precision cannot be provided by the publicly available elements in GP/TLE format. However, the information is available to the spacecraft operators, possibly with an even higher precision.

We therefore recommend that the orbital parameters (suited to high fidelity models, with covariance/uncertainty information) be reliably made available by the satellite operators to the observatories. Should the full precision orbital data be commercially sensitive information, they could of course be degraded/truncated, to a precision allowing for sub-arcminute uncertainty on the position. The detailed exchange of requirements resulting in such an arrangement can be part of the ongoing dialogue and cooperation between the astronomical community and the space industry.

### 3.2.3.2 Accuracy, maneuvers, and operator data

Most orbital data, including TLEs, have no accuracy or uncertainty information. Some orbital solution data formats allow provision of uncertainties in the form of a time series of position-velocity  $6 \times 6$  covariance matrices.

Satellite operators regularly perform maneuvers to maintain or change their orbits (e.g, ESA performs weekly manoeuvres — “burns” — for its Earth observation satellites; and SpaceX adjusts the orbits of its Starlink satellites frequently for orbit raising and slot relocation). These maneuvers cannot be predicted by external observers and, of course, make existing orbit predictions obsolete. There is now a mechanism by which operators can inform 18SPCS of planned trajectories and maneuvers and allow 18SPCS to make these predictions available to other operators (but not to the public). In addition to incorporating planned burns, operator orbital predictions are often based on on-board GNSS receivers, are more accurate than passive tracking by surveillance systems, and may include covariance data.

GP data are given in ASCII TLE 80-characters-per-line format inherited from punch card days; they are also available in JSON and other representations. Ephemeris data formats include: CCSDS Orbit Ephemeris Message (OEM), NASA ITC, and ILRS CPF. CPF is used by the satellite laser ranging community who often prefer the ITRF frame. More details of the formats are given in the appendix.

### 3.2.4 PassPredict simple mode: detailed requirements for inputs

- Observatory location. For a ground based observatory, the station location shall be provided either in a geodetic frame using latitude, longitude and height (e.g., in degrees and meters) or in a geocentric Earth-fixed cartesian frame (ITRF) using x,y,z coordinates. For space-based telescopes, an ephemeris is required. For compatibility with the ground-based case, this should also be in geodetic or ITRF frame versus time. Alternatively we may want to support doing all the

<sup>6</sup> <https://celestrak.com/NORAD/elements/supplemental>

<sup>7</sup> <https://space-track.org> (under 'Public Files')

calculations in inertial TEME or ICRS frames. The exposure times of space observatories are typically very long compared to the time it takes for the observatory position to change significantly.

- Observation schedule.
- The centre pointing of the telescope during exposure shall be defined with azimuth and elevation or topocentric ICRS (or TEME) right ascension and declination.
- The field of view shall be defined using a radius in degrees in the case of a circular shape, horizontal and vertical dimensions for a rectangular shape or using a polygon to allow arbitrary shapes
- The pixel size shall be given in arcseconds.

### 3.2.5 PassPredict: algorithm

We can search for passes using a brute force grid search with fixed time steps, or by using a root search algorithm (Oltrogge, Kelso & Seago, 2011). Batch or individual requests could be generated by existing tools which support observation requests, but they would need updating to provide an interface.

For each satellite, we calculate the position at a series of time steps specified by the user. For each step, we perform the following steps:

1. Calculate the geocentric satellite position and uncertainty (interpolating state and covariance from the ephemeris or propagating TLE and using a fixed uncertainty estimate). Note that covariance interpolation can lead to unphysical results (results which are non-positive-definite or have negative variance). See the astroplan package (<https://astronplan.readthedocs.io/en/latest/>) for approaches to handling this.
2. Calculate the difference vector between the observatory location and the satellite position.
3. Transform this to a unit vector (the “line of sight”) in the observatory topocentric frame.
4. Determine the telescope field of view at this time in the topocentric frame.
  - a. There remain detailed issues to resolve: how to handle aberration and refraction; whether we convert the FOV from apparent to true RA/Dec, or conversely convert the satellite vectors from true to apparent. An observing schedule is normally given in true RA/Dec rather than apparent, so the former approach seems better.
  - b. If the line of sight is within the field of view, the satellite is geometrically observable (Alfano, Negron & Moore, 1992).
5. Calculate the satellite topocentric angular velocity in the instrument frame.
  - a. This is not necessarily tracking at the sidereal rate; however, this is unlikely to be a significant issue.
6. Calculate the geocenter-Sun vector from, e.g., the JPL ephemeris.
7. Subtract the satellite-geocenter vector to obtain the satellite-Sun unit vector.
8. Compare the geocenter-Sun vector and Earth radius to see if the Sun is above the horizon as seen by satellite — if so, it is illuminated.

In simple mode:

1. Output the result line if the satellite is geometrically observable.
2. Keep track of whether the satellite was observable at the previous time step; if so, it’s the same streak and has the same streak ID (identification) number. If not, increment the streak ID number.
3. The result line indicates: satellite catalog number; streak ID number; predicted topocentric right ascension and declination at the time step; a flag to say whether the satellite is illuminated or not; the location of the streak in the field of view.

In advanced mode:

1. Use the BRDF (and attitude model if available) to determine the predicted magnitude in specified bands. Add to the output line.
2. Use the angular velocity and pixel size to predict the surface brightness of the streak per pixel. Add to the output line.

### 3.2.6 Available software for PassPredict simple mode

*Table 1: The table below identifies a variety of existing software that may have capabilities relevant to the PassPredict effort.*

	Package	Capabilities
<b>Main candidates for orbit calculations</b>	GMAT (NASA)	TLE propagation, OEM interpolation, frame conversion, position prediction (GMAT, n.d.)
	Orekit	OEM interpolation, frame conversion. The code is from a French company (CSGroup). OreKIT is used in Moriba Jah's Orbit DetPy (Iyer, 2019).
	OrbitDetPy	Wrapper on OreKIT with useful functionality, from UT Austin group <a href="https://github.com/ut-astris/orbdepy">https://github.com/ut-astris/orbdepy</a> (Iyer, 2019) and <a href="https://github.com/IBM/arcade">https://github.com/IBM/arcade</a> (IBM/U of Texas).
<b>Wrappers for observation requests</b>	Astropy.astroplan	Observation planning
	TOM Toolkit	Observation planning (Street et al 2018)
<b>Other packages of interest</b>	STK	AGI; not free or open source
	Freeflyer	A.i.-solutions; not free or open source
	ERFA	Astropy; frame conversion routines
	Slrfield	Satellite laser ranging pass predictions <a href="https://pypi.org/project/slrfield">https://pypi.org/project/slrfield</a>
	PROOF	ESA; uses GP/TLE to compute individual topocentric pass predictions with configurable observatory parameters, including instrument field of view, etc. It is an open source Fortran code with a Java interface.
	Monte	JPL (not open source)

Notes: STK and Freeflyer are widely-used industry-standard packages of particular note. R. Street (Las Cumbres Observatory) noted an interest in building a component to the LCO TOM Toolkit to wrap the U. Texas OrbitDetPy work.

### 3.2.7 PassPredict advanced mode — considerations

In order to predict the apparent brightness of satellites a model for the reflectance distribution must be provided. The brightness can be predicted in a deterministic way using a satellite model or some

approximation (e.g., a look-up table) assuming knowledge of the attitude state. Operators could be encouraged to share their attitude states using, e.g., the CCSDS ADM formats (similarly to ephemeris files). Many operators also follow some attitude law which could be considered in the code. Alternatively, the brightness can be bounded statistically irrespective of the attitude state, e.g. based solely on Sun-phase angle.

It would be helpful if operators made available BRDFs, satellite models, and attitude control profiles to allow us to make detailed brightness predictions.

**Effort required.** We have not made a level of effort estimate for the PassPredict work. The existing relevant software suggests that the actual software development will be a smaller job than TrailMask; however, a robust system to manage and interface with the ephemeris data will likely take significant and continuing resource investment.

## 3.3 SATCON1 recommendation 3: Simulation Tools

*Support selected detailed simulations of the effects on data analysis systematics and data reduction signal-to-noise impacts of masked trails on scientific programs affected by satellite constellations. Aggregation of results should identify any lower thresholds for the brightness or rate of occurrence of satellite trails that would significantly reduce their negative impact on the observations.*

### 3.3.1 EphemSimulate

To model the effects of future constellations, we need to be able to go from a constellation description to a simulated ephemeris database; EphemSimulate would do this.

Inputs:

- A. Constellation shell parameters
- B. Observation date

Outputs

- 1. Simulated ephemeris database

A typical constellation description defines a number of layers with fixed altitude and inclination (see Fig. 6 for an example). Each layer specifies the number of orbital planes and number of satellites per plane. We can assume that the planes are evenly spaced and that the satellites in a single plane are on average evenly spaced along the orbit (possibly with some rule for adding some randomness to the phases along the orbit). This allows us to instantiate a suitable set of orbital elements for each satellite in the constellation. For this purpose (to assess the impact of a particular new constellation design), perfect circular Keplerian orbits are likely a sufficiently accurate representation; detailed propagation models are not needed.

Since the deployment of Starlink it has become clear that satellites may spend a significant fraction of their lifetime in ascent and descent orbits and in plane-adjusting drift orbits at intermediate altitudes, so simulations may want to include these effects as well.

## MODEL VII: Chinese GW Constellation (2021 revision, 12992 satellites)

Layer	Element	Altitude (km)	Inclination (deg)	No of planes	Sats per plane	Total sats
A	1	590	85.0	16	30	480
A	2	600	50.0	40	50	2000
A	3	508	55.0	60	60	3600
B	4	1145	30.0	48	36	1728
B	5	1145	40.0	48	36	1728
B	6	1145	50.0	48	36	1728
B	7	1145	60.0	48	36	1728

Figure 6. Typical example constellation definition, in this case for a proposed Chinese constellation.

The output of this tool could then be passed to PassPredict or a similar algorithm to model the observability of the constellation at a given observatory and date. In an advanced mode, we should also predict the satellite brightness. Simulations should include cases for professional observations and also of the effect on the naked-eye sky. As to existing software, various unpublished research codes exist but would need to be modified to generate the ephemeris database output and improved to be robust enough for general use.

### 3.3.1.1 PassProbability

We can use the approach of EphemSimulate in a statistical mode to predict the average density of satellite passes on the sky in a given situation. We call this tool PassProbability. This can be used for long-term impact studies or as a quick alternative to the computation-intensive PassPredict to estimate the probability that a given observation will be affected by a satellite trail. Bassa et al. (2021, in preparation) have generated an analytic satellite density model that can be used to go from the constellation definitions to probability sky maps and calendars. The probability that an observation will be affected varies by orders of magnitude with only slight changes in the observation parameters.

We can consider two modes of output:

1. A Probability Map (Figure 7) which shows the fraction of exposure lost due to satellite trails as a function of celestial position. A value greater than 1 indicates that the exposure is entirely lost.
2. A Probability Calendar (Figure 8). Here we calculate the probability of exposure loss for a specific target (in this case, the Large Magellanic Cloud) as a function of time of night and observation date. The idea is to plan the optimal dates for an observing run. The solar elevation is indicated by the blue contours; yellow-green slanted lines indicate the elevation of the target, with elevations below 20 degrees considered unobservable and shaded in gray. The color scale indicates the expected fraction of exposures lost.



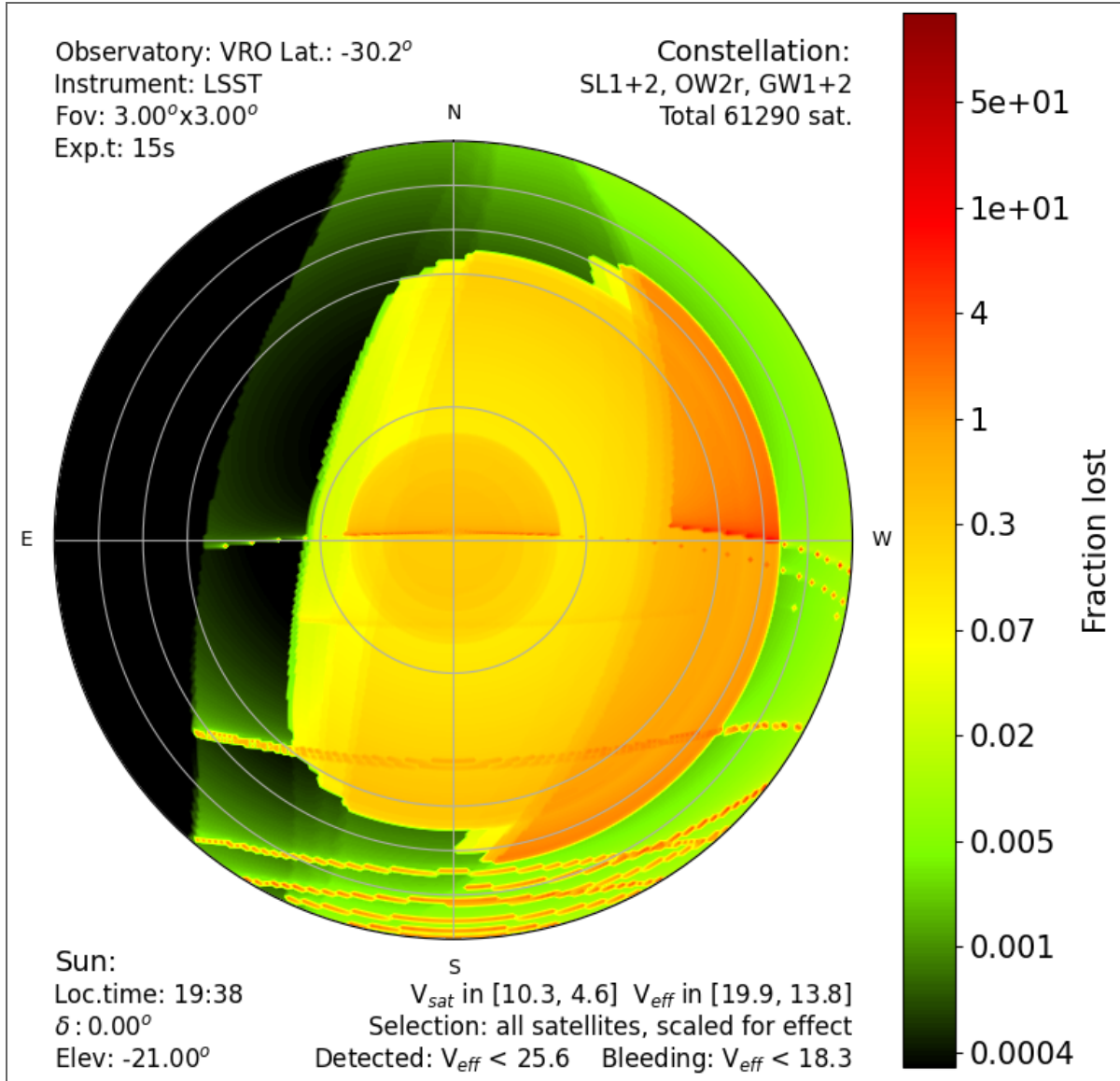


Figure 7. An example probability map displaying fraction of exposure lost due to satellite trails as a function of celestial position.



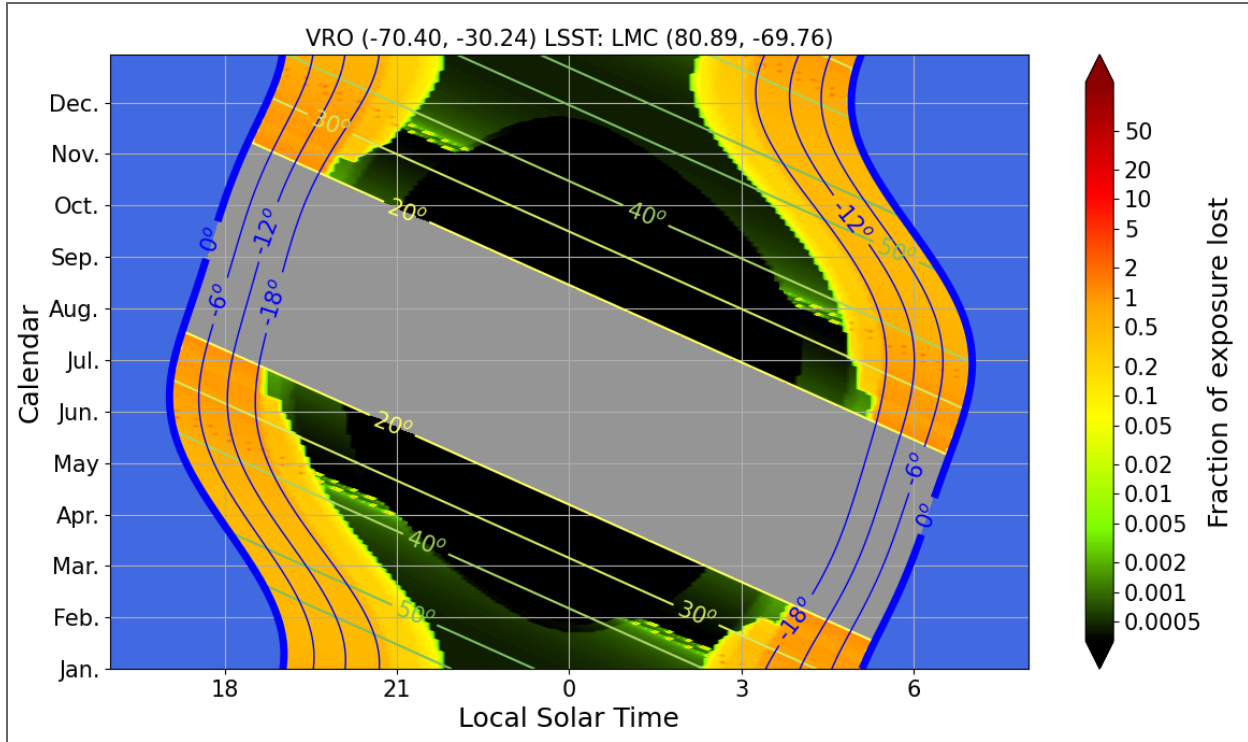


Figure 8. A Probability Calendar displaying the probability of exposure loss for a specific target (in this case, the Large Magellanic Cloud) as a function of time of night and observation date.

### 3.3.2 ImageSimulate

We need to provide base images to add simulated trails to. One could argue that you can just use existing images and no basic image simulation tool is needed. However, it is useful to have the capability to create images with known (because simulated) source properties (e.g., faint sources with a variety of known magnitudes) to properly assess detection thresholds.

Inputs:

- Observation parameters, the same as fed to PassPredict
- Base image (e.g., from archive for desired telescope)
- List of additional sources to simulate with a point spread function and/or thumbnail image

Output:

- Simulated image (no trails, but with added test sources)

Existing software:

- Skymaker, from the Astromatic package (the software for the Canada France Hawai'i Telescope's MegaCam; also the same team as SExtractor, SCamp etc.)
- GalSim (GalSim-developers, 2012) is a very widely used tool in wide-field optical cosmology
- The official simulation suite for the LSST Dark Energy Science Collaboration is DC2. It actually uses GalSim under the hood. (LSST Dark Energy Science Collaboration, 2021)

GalSim is an open-source software library to perform image simulations. It can be used either through its python interface, or as an executable, configurable through YAML files. Most of the computation-heavy parts of the code are written in C++ for performance. In its simplest configuration, GalSim uses simple

parametric models for both galaxies (e.g., Sersic, exponential) and point spread functions (e.g., Moffat). The former can also be generated from real Hubble Space Telescope images (provided the instrument for which images are to be simulated has a larger point spread function). The latter can be created as the convolution of an optical and an atmospheric point spread function. The optical part can be simulated if the user provides a set of instrument related parameters. Alternatively, an external calibration image of the point spread function can be used. The positions and parameters of the objects to be simulated can be read from a catalog. Several options also exist for noise, detector effects and WCS. The simulated image outputs are usually stored in FITS files. At present, it does not handle simulation of several image artefacts, including trails due to satellites.

### 3.3.3 TrailSimulate

To assess impacts of satellite trails on science, we need to create simulated images with specific trail properties. This tool will require detailed (probably instrument-dependent, plugin) models of the appearance of trails and related instrumental effects.

A first mode of the tool would be driven by a specific satellite pass prediction.

A second mode might be to give the code a trail occurrence distribution as a function of brightness (e.g., 10 trails per square degree per hour uniformly distributed between magnitudes 4 and 6) rather than a PassPredict output. We will refer to this as the “rate input mode”.

Inputs:

- A. Transit list from PassPredict run on output from EphemSimulate or (rate input mode) rate of trails as a function of brightness
- B. Observation parameters, the same as fed to PassPredict
- C. Simulated image without trails
  - a. Must be consistent with B.
- D. Model (code) for generating trails including CCD and optical side effects

Outputs:

1. Simulated image with trails
2. Fraction of image pixels affected (including by side effects)

**Existing software.** We are not aware of any existing software that would address this issue.

### 3.3.4 TrailAssess

We need to assess what the scientific impact of trails has been on our images. One way to do this is to take two images of the same field, one unaffected and one affected (either with trails, or still degraded after trail removal). These could be real images (with different epochs), or a pair of simulated images with and without trails. One metric of the effect on science is to detect and parameterize sources in the field, and compare the derived source list and source parameters before and after degrading the image with trails.

Inputs:

- A. Simulated image without trails, output from ImageSimulate
- B. Simulated image with trails, output from TrailSimulate or from TrailMask (with trails removed at some level)
- C. Trail catalog (from TrailMask)

D. Data reduction parameters, to be determined

Outputs:

1. Point source detection list with source parameters for both images
2. Extended source detection list with source parameters for both images
3. Detection efficiency and photometric accuracy for both lists
4. Sensitivity limit for both lists
5. Derived output: percentage degradation in source detection efficiency vs brightness percentage degradation in detection threshold.

### 3.3.5 Simulation assessment

This is not a software application per se. We are also tasked with aggregating the simulation results generated by the tools described in the preceding subsection. and interpreting them, summarizing them for the community.

We will need to define a set of simulations to cover the relevant parameter spaces and provide sufficient data for an assessment, then actually run the simulations. Then we need to generate summary trend plots and tables versus time of year, observatory location, etc., for different types of observation/science, different telescopes, and for different constellation scenarios. These will allow us to provide recommendations on desired limits on trails and make progress on suggesting corresponding limits on various types of satellite.

## 4. Implementation Timescale

Specifying an exact timetable for the implementation of the software tools described above lies outside the capabilities of this Working Group, and the conceptual outline of the packages is not yet sufficiently detailed to allow credible planning of implementation timescales. However, the Working Group emphasizes the importance of these tools' being available to the astronomical community on a timescale commensurate with that of the development of the satellite constellations; the tools need to be available before the need for them is so pressing that observations are severely disrupted. Bearing in mind that software development projects frequently take significantly longer than originally expected, the Working Group stresses the urgent need to invest in the development of these tools.

## 5. Cross-Group Coordination

We will need to continue coordination with the Observations Working Group on the issue of publicly accessible satellite positional information, which is a key input for some of the above software. The Policy Working Group could argue for national policies to support the availability of the needed inputs.

We welcome further input from the Community Engagement Working Group about what stakeholders, if any, will need to access the software beyond the professional and hobbyist observer community, and what implications that has for the interfaces.

# 6. Conclusions

We summarize our findings in the following conclusions:

1. We re-emphasize SATCON1 recommendations 1 to 3. New tools are critical to partially mitigate the constellation impacts on astronomy. The PassPredict software will allow astronomers to determine which observations may be affected and in conjunction with simulations may allow quantification of the degradation of science data expected in a particular situation. The TrailMask software will allow some science to be salvaged from some affected datasets and reduce the chance of spurious results being published. A large simulation and modeling effort will allow the community to assess impacts of current and future constellations on both ground- and space-based observations and establish recommended constraints on the design of constellations.
2. In the report we have provided a moderately detailed analysis of requirements, interfaces and algorithms that may serve as a starting point for software implementation.
3. Some software already exists to help with parts of these tasks. However, much of it is specialized to particular instruments or situations, and needs to be generalized.
4. There are gaps where no suitable software exists, and a significant software development effort is warranted. Project management, documentation, user support and maintenance will all be important and will require substantial resources and funding. Educational materials (e.g., lesson plans) are also desirable.
5. To support the diverse community of night-sky users, software must be provided in several forms: libraries (integrated with core astronomy interfaces like the Astropy project), applications for data pipelines, web services and planetarium-compatible services.
6. We conclude that there is an urgent need to develop a set of test cases, including example datasets covering a wide range of instrument and satellite-trail properties which can serve as a standard test suite for the development of the software and as benchmark comparisons for both archival and new sources of data.
7. We endorse the SatHub concept developed by the Observations Working Group. SatHub provides a natural home for curated software (and links to external software), satellite catalog and ephemeris access, test data, and documentation. This aspect of SatHub, like the others, will need continuing development, support and maintenance at a professional level.
8. The constellations are being launched now but software takes time to develop. Resources should be made available as soon as possible.
9. If the satellite constellations are deployed as planned we find that no software solution can fully mitigate the impact on astronomical observations. The problems with spectroscopic observations are particularly hard to solve. It is likely that many ground-based observatories around the world will be forced to make a significant investment in hardware such as auxiliary spotting cameras or other solutions to deal with the problem of satellite streaks. However, the effects of satellite constellations on a given observatory will depend on specifics such as aperture, etendue, pixel size, observing strategy, and other factors; for some the impact will be small and mitigations will not be required, while for others the impact will be serious. Each observatory will need to make its own assessment.

# References

- 18th Space Control Squadron (2020). *Spaceflight Safety Handbook for Operators*.  
[https://www.space-track.org/documents/Spaceflight\\_Safety\\_Handbook\\_for\\_Operators.pdf](https://www.space-track.org/documents/Spaceflight_Safety_Handbook_for_Operators.pdf)
- AGI. (n.d.) *STK SatPro*. Agi.com. <https://www.agi.com/products/stk-specialized-modules/stk-satpro>
- a.i. solutions. (n.d.) *Freeflyer Astrodynamics Software*. ai-solutions.com.  
<https://ai-solutions.com/freeflyer-astrodynamic-software/>
- Alfano, S., Negron Jr, D., & Moore, J. L. (1992). *Rapid determination of satellite visibility periods*. Journal of the Astronautical Sciences, 40(2), 281
- Astroplan. [Computer Software].(n.d.). Retrieved from <https://astroplan.readthedocs.io/en/latest/>
- CCSDS. (2009). *Orbit Data Messages*. CCSDS 502.0-B-2, Blue Book,  
<https://public.ccsds.org/Pubs/502x0b2c1e2.pdf>
- CS-Group (2002). *Orekit: An accurate and efficient core layer for space flight dynamics applications*. Orekit.org. <https://www.orekit.org/>
- Desai, S., Mohr, J. J., Bertin, E., Kümmel, M. & Wetzstein, M. (2016). *Detection and removal of artifacts in astronomical images*. Astronomy and Computing. 16. 67. doi:10.1016/j.ascom.2016.04.002
- GalSim-developers. (2012). *GalSim-developers/GalSim*. Github.com.  
<https://github.com/GalSim-developers/GalSim>
- GMAT: General Mission Analysis Tools. (n.d.). GMAT Wiki. Retrieved September 27, 2021, from  
<https://gmat.atlassian.net/wiki/spaces/GW/overview?mode=global>
- Gruen, D., Seitz, S. & Bernstein, G. M. (2014). *Implementation of Robust Image Artifact Removal in SWarp through Clipped Mean Stacking*. Publications of the Astronomical Society of the Pacific, 126, 158. doi:10.1086/675080
- ILRS Data Format Working Group (2006). *Consolidated Laser Ranging Prediction Format*. Version 1.01,  
[https://ilrs.gsfc.nasa.gov/docs/2006/cpf\\_1.01.pdf](https://ilrs.gsfc.nasa.gov/docs/2006/cpf_1.01.pdf)
- International Business Machines. (n.d.) *IBM/arcade*. Github.com. <https://github.com/IBM/arcade>
- Iyer, S. (2019) *ut-astris/orbdepy*. Github.com. <https://github.com/ut-astris/orbdepy>
- Kelso, T. S., (n.d.). *Supplemental Two-Line Element Sets*. CelesTrak.com.  
<https://celestrak.com/NORAD/elements/supplemental>
- LSST Dark Energy Science Collaboration (LSST DESC) et al. (2021). *The LSST DESC DC2 Simulated Sky Survey*. Astrophysical Journal Supplement, 253, 1, 31. doi:10.3847/1538-4365/abd62c
- McCant, M., (n.d.). *Mike McCants' Satellite Tracking TLE ZIP Files*. Prismnet.com.  
<https://www.prismnet.com/~mmccants/tles/index.html>

- Oltrogge, D., Kelso, T. S. & Seago, J. (2011). Ephemeris requirements for Space Situational Awareness. 140. AAS 11-151.  
[https://comspoc.com/Resources/Papers/20110215\\_Ephem\\_Rqmts\\_for\\_SSA\\_Oltrogge\\_Kelso\\_AAS\\_11-151\\_SUBMITTED.pdf](https://comspoc.com/Resources/Papers/20110215_Ephem_Rqmts_for_SSA_Oltrogge_Kelso_AAS_11-151_SUBMITTED.pdf)
- Science Applications International Corporation. (2021, 27 September). Space-Track.org.  
<https://space-track.org>
- SkyMaker. (2006). Astromatic.net Retrieved on September 27, 2021 from  
<https://www.astromatic.net/software/skymaker/>
- Street, R. A., Bowman, M., Saunders, E. S. & Boroson, T. (2018). *General-purpose software for managing astronomical observing programs in the LSST era*. SPIE. 10707, 1070711. doi:10.1117/12.2312293
- Teimoorinia, H. , Shishehchi, S., Tazwar, A. , Lin, P., Archinuk, F., Gwyn, S. D. J. & Kavelaars, J. J., (2021), *An Astronomical Image Content-based Recommendation System Using Combined Deep Learning Models in a Fully Unsupervised Mode*. The Astronomical Journal, 161, 227.  
<https://doi.org/10.3847/1538-3881/abea7e>
- Vallado, D. A. (2001). *Fundamentals of astrodynamics and applications* (Vol. 12). Springer Science & Business Media.
- Virtanen, J., Poikonen, J., Sääntti, T., Komulainen, T., Torppa, J., Granvik, M., Muinonen, K., Pentikäinen, H., Martikainen, J., Näränen, J. & Lehti, J. (2016). *Streak detection and analysis pipeline for space-debris optical images*. Advances in Space Research, 57(8), 1607. doi: 10.1016/j.asr.2015.09.024
- Walker, C., Hall, J., Allen, L., Green, R., Seitzer, P., Tyson, A., Bauer, A., Krafton, K., Lowenthal, J., Parriott, J., Puxley, P., Abbott, T., Bakos, G., Barentine, J., Bassa, C., Blakeslee, J., Bradshaw, A., Cooke, J., Devost, D.,...Yoachim, P. (2020). *Impact of Satellite Constellations on Optical Astronomy and Recommendations Towards Mitigations*. NSF's NOIRLab.  
<https://noirlab.edu/public/products/techdocs/techdoc003/>
- Xu, C., McCully, C., Dong, B., Howell, D. A., & Sen, P. (2021). *Cosmic-CoNN: A Cosmic Ray Detection Deep-Learning Framework, Dataset, and Toolkit*. arXiv:2106.14922  
<https://arxiv.org/abs/2106.14922>

# Appendix A: Ephemeris file formats

## 1.1 File formats

Common ephemeris formats are CCSDS OEM, NASA ITC format, or ILRS CPF. They provide additional information such as the used frame (e.g., EME2000, TEME, ITRF), interpolation method and order, and manoeuvre epochs. ILRS CPF files are used within the satellite laser ranging community for station predictions. Operators of laser ranging stations often prefer the ITRF frame to simplify the computation (as there is no need for any frame conversions).

### 1.1.1 Example CCSDS OEM

File extracted from the standard.

```
CCSDS_OEM_VERSION = 2.0
CREATION_DATE = 1996-11-04T17:22:31
ORIGINATOR = NASA/JPL

META_START OBJECT_NAME = MARS GLOBAL SURVEYOR
OBJECT_ID = 1996-062A
CENTER_NAME = MARS BARYCENTER
REF_FRAME = EME2000
TIME_SYSTEM = UTC
START_TIME = 1996-12-28T21:29:07.267
USEABLE_START_TIME = 1996-12-28T22:00:02.5
USEABLE_STOP_TIME = 1996-12-30T01:18:02.5
STOP_TIME = 1996-12-30T01:28:02.267
INTERPOLATION = HERMITE
INTERPOLATION_DEGREE = 7
META_STOP

1996-12-28T21:29:07.267 -2432.166 -063.042 1742.754 7.33702 -3.495867 -1.041945
1996-12-28T21:59:02.267 -2445.234 -878.141 1873.073 1.86043 -3.421256 -0.996366
1996-12-28T22:00:02.267 -2458.079 -683.858 2007.684 6.36786 -3.339563 -0.946654
...
1996-12-30T01:28:02.267 2164.375 1115.811 -688.131 -3.53328 -2.88452 0.88535

COVARIANCE_START

EPOCH = 1996-12-28T21:29:07.267
COV_REF_FRAME = EME2000
3.3313494e-04
4.6189273e-04 6.7824216e-04
-3.0700078e-04 -4.2212341e-04 3.2319319e-04
-3.3493650e-07 -4.6860842e-07 2.4849495e-07 4.2960228e-10
-2.2118325e-07 -2.8641868e-07 1.7980986e-07 2.6088992e-10 1.7675147e-10
-3.0413460e-07 -4.9894969e-07 3.5403109e-07 1.8692631e-10 1.0088625e-10 6.2244443e-10

EPOCH = 1996-12-29T21:00:00.000
COV_REF_FRAME = EME2000
3.4424505e-04
4.5078162e-04 6.8935327e-04
-3.0600067e-04 -4.1101230e-04 3.3420420e-04
-3.2382549e-07 -4.5750731e-07 2.3738384e-07 4.3071339e-10
-2.1007214e-07 -2.7530757e-07 1.6870875e-07 2.5077881e-10 1.8786258e-10
-3.0302350e-07 -4.8783858e-07 3.4302008e-07 1.7581520e-10 1.0077514e-10 6.2244443e-10

...

COVARIANCE_STOP
```



## 1.1.2 Example CPF

File extracted from the format description.

```
H1 CPF 1 AIU 2005 11 16 4 8201 gps35
H2 9305401 3535 22779 2005 11 15 23 59 47 2005 11 20 23 29 47 900 1 1 0 0 0
H9
10 0 53689 86387.000000 0 -13785362.868 -12150743.695 19043830.747
10 0 53690 887.000000 0 -13656536.158 -14288496.731 17628980.237
10 0 53690 1787.000000 0 -13618594.073 -16250413.260 15908160.431
10 0 53690 2687.000000 0 -13647177.924 -18001187.561 13911910.138
10 0 53690 3587.000000 0 -13712868.344 -19511986.614 11675401.577
10 0 53690 4487.000000 0 -13782475.931 -20761369.576 9237779.852
...
99
```

## 1.1.3 Example NASA Modified ITC Ephemeris format

Dummy file content.

```
created:2021-01-01 01:00:00 UTC
ephemeris_start:2021-01-01 02:00:00 UTC ephemeris_stop:2021-05-03 02:00:00 UTC step_size:60
ephemeris_source:blend
UNW
202100102000.000 -5000.000000000 1000.000000000 3000.000000000 1.500000000 -5.000000000 4.000000000
1.000000000e-01 -1.000000000e-02 1.000000000e-03 -1.000000000e-04 -1.000000000e-05 1.000000000e-06 1.000000000e-06
-1.000000000e-07 1.000000000e-08 1.000000000e-09 -1.000000000e-10 1.000000000e-11 -1.000000000e-12 -1.000000000e-13
1.000000000e-14 -1.000000000e-15 1.000000000e-16 1.000000000e-17 1.000000000e-18 -1.000000000e-19 1.000000000e-20
2021001020100.000 -5200.000000000 700.000000000 3200.000000000 1.700000000 -5.300000000 4.200000000
1.000000000e-01 -1.000000000e-02 1.000000000e-03 -1.000000000e-04 -1.000000000e-05 1.000000000e-06 1.000000000e-06
-1.000000000e-07 1.000000000e-08 1.000000000e-09 -1.000000000e-10 1.000000000e-11 -1.000000000e-12 -1.000000000e-13
1.000000000e-14 -1.000000000e-15 1.000000000e-16 1.000000000e-17 1.000000000e-18 -1.000000000e-19 1.000000000e-20
...
```

# Appendix B: Glossary of Abbreviations

Abbrev.	Full Title	Type	URL for further information
18SPCS	18th Space Control Squadron, US Space Force	Organization	<a href="https://en.wikipedia.org/wiki/18th_Space_Control_Squadron">https://en.wikipedia.org/wiki/18th_Space_Control_Squadron</a>
ADM	-	Satellite data format	
API	Applications Programming Interface	Software term	
BRDF	Bidirectional Reflectance Distribution Function	Optics term	
CADC	Canadian Astronomy Data Center	Organization	<a href="https://www.cadc-ccda.hia-ihp.nrc-cnrc.gc.ca/en/">https://www.cadc-ccda.hia-ihp.nrc-cnrc.gc.ca/en/</a>
CCD	Charge coupled device	Astronomical detector	
CCSDS		Orbit data format	<a href="https://public.ccsds.org">https://public.ccsds.org</a>
CPF	-	Orbit data format	<a href="https://ilrs.gsfc.nasa.gov/data_and_products/formats/cpf.html">https://ilrs.gsfc.nasa.gov/data_and_products/formats/cpf.html</a>
CNN	Convolutional Neural Network	Algorithm	<a href="https://en.wikipedia.org/wiki/Convolutional_neural_network">https://en.wikipedia.org/wiki/Convolutional_neural_network</a>
EME2000	Earth Mean Equator 2000	Coordinate System	<a href="https://en.wikipedia.org/wiki/Earth-centered_inertial">https://en.wikipedia.org/wiki/Earth-centered_inertial</a>
ESA	European Space Agency	Organization	<a href="https://esa.org">https://esa.org</a>
FOV	Field of view	Astronomy term	
FTE	Full Time Equivalent	Management term	<a href="https://en.wikipedia.org/wiki/Full-time_equivalent">https://en.wikipedia.org/wiki/Full-time_equivalent</a>
GEO	Geosynchronous Earth Orbit	Orbit category	<a href="https://en.wikipedia.org/wiki/Geostationary_orbit">https://en.wikipedia.org/wiki/Geostationary_orbit</a>
GNSS	Global Navigation Satellite System	Navigation term	
GP	General Perturbations	Orbit theory	
GPU	Graphics Processing Unit	Computer hardware	<a href="https://en.wikipedia.org/wiki/Graphics_processing_unit">https://en.wikipedia.org/wiki/Graphics_processing_unit</a>

ICRS	International Celestial Reference System	Coordinate system	<a href="https://en.wikipedia.org/wiki/International_Celestial_Reference_System_and_Frame">https://en.wikipedia.org/wiki/International_Celestial_Reference_System_and_Frame</a>
IFU	Integral Field Unit	Astronomical detector	<a href="https://en.wikipedia.org/wiki/Integral_field_spectrograph">https://en.wikipedia.org/wiki/Integral_field_spectrograph</a>
ILRS	International Laser Ranging Service	Organization	
ITC	International Telecommunications Corporation	Orbit data format	<a href="https://platform.leolabs.space/documentation/ephemerides_file_formats">https://platform.leolabs.space/documentation/ephemerides_file_formats</a>
ITRF	International Terrestrial Reference Frame	Coordinate system	
IVOA	International Virtual Observatory Alliance	Organization	<a href="https://ivoa.net">https://ivoa.net</a>
JPL	Jet Propulsion Laboratory	Organization	<a href="https://www.jpl.nasa.gov/">https://www.jpl.nasa.gov/</a>
JSON	-	Software protocol	<a href="https://www.json.org">https://www.json.org</a>
LEO	Low Earth Orbit	Orbit category	<a href="https://en.wikipedia.org/wiki/Low_Earth_orbit">https://en.wikipedia.org/wiki/Low_Earth_orbit</a>
LSST	Legacy Survey of Space and Time	Astronomical survey	<a href="https://www.lsst.org">https://www.lsst.org</a>
MEO	Medium Earth Orbit	Orbit category	<a href="https://en.wikipedia.org/wiki/Medium_Earth_orbit">https://en.wikipedia.org/wiki/Medium_Earth_orbit</a>
NASA	National Aeronautics and Space Administration	Organization	<a href="https://nasa.gov">https://nasa.gov</a>
OEM	Orbit Ephemeris Message	Orbit data format	
pyVO	-	Software protocol	<a href="https://pyvo.readthedocs.io">https://pyvo.readthedocs.io</a>
S/N	Signal-to-noise	Data analysis	
SKKP	Tsentr Kontroly'a Kosmicheskya Prostranstva	Organization	<a href="https://en.wikipedia.org/wiki/Main_Centre_for_Reconnaissance_of_Situation_in_Space">https://en.wikipedia.org/wiki/Main_Centre_for_Reconnaissance_of_Situation_in_Space</a>
SP	Special Perturbations	Orbit theory	
TAP	Table Access Protocol	Software protocol	<a href="https://www.ivoa.net/documents/TAP/">https://www.ivoa.net/documents/TAP/</a>
TEME	True Equator Mean Equinox	Coordinate System	<a href="https://docs.astropy.org/en/stable/api/astropy.coordinates.builtin_frames.TEME.html">https://docs.astropy.org/en/stable/api/astropy.coordinates.builtin_frames.TEME.html</a>

TLE	Two Line Elements	Orbit data format	
WCS	World Coordinate System	Software protocol	<a href="https://www.atnf.csiro.au/people/mcalabre/WCS/">https://www.atnf.csiro.au/people/mcalabre/WCS/</a>
WGS84	World Geodetic System 1984	Geodetic frame	<a href="https://gisgeography.com/wgs84-world-geodetic-system/">https://gisgeography.com/wgs84-world-geodetic-system/</a>
WWT	World Wide Telescope	Web application	<a href="https://worldwidetelescope.org/webclient/">https://worldwidetelescope.org/webclient/</a>
YAML	-	Software language	<a href="https://yaml.org">https://yaml.org</a>