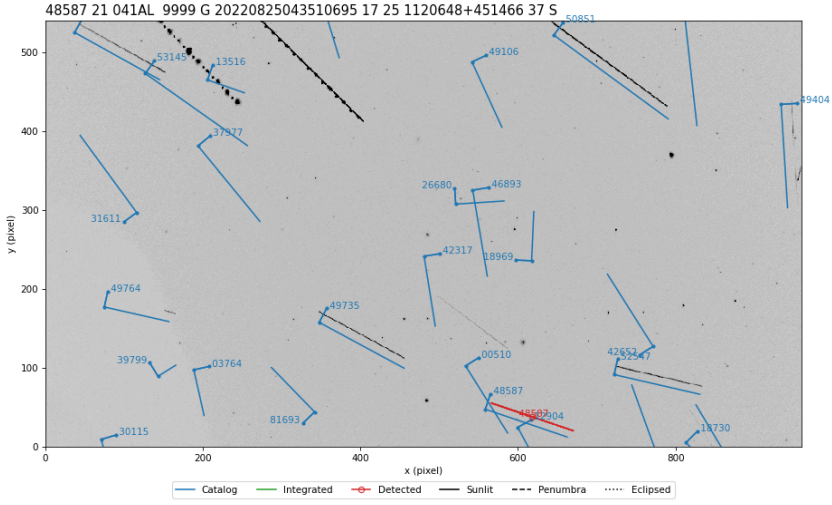
# SatNOGS Optical Unofficial Guide

DRAFT

ISBN 978-1-387-65419-2

UT Date: 2022-08-25T04:35:05.706 COSPAR ID: 9999
R.A.: 174.602042 (17.2") Decl.: 43.540431 (22.6")
FOV: 12.86° x7.24° Scale: 48.23"x48.25" pix$^{-1}$

48587 21 041AL 9999 G 20220825043510695 17 25 1120648+451466 37 S



# SatNOGS Optical

*Unofficial*

# Guide

**Jeff Moe**

*Loveland, Colorado, USA*

2022

**SatNOGS Optical Unofficial Guide**

by Jeff Moe

Copyright © 2022 Jeff Moe.

Permission is granted to copy, distribute and/or modify this document under the terms of the Creative Commons Attribution 4.0 International Public License (CC BY-SA 4.0).

Published by Jeff Moe, Loveland, Colorado, USA.

ISBN: 978-1-387-65419-2 XXX DRAFT NOT FINAL

10 9 8 7 6 5 4 3 2 1 0 20220909

Dedicated to free users

*Space should be claimed the libre way.*

---

Libre Space Foundation

# Contents

**List of Figures** . . . . . **viii**

**List of Tables** . . . . . **x**

*Introduction*

**SatNOGS Optical** . . . . . **xi**

Unofficial Guide Overview . . . . . xii

Libre Space Foundation . . . . . xiii

SatNOGS Network . . . . . xiii

Optical Network Status . . . . . xiii

**1** *Satellites*

**Artificial Satellites in Space** . . . . . **15**

1.1 Overview of Satellites . . . . . 16

1.2 SatNOGS DB . . . . . 16

1.3 Libre Satellites . . . . . 17

1.4 Satellite Orbits . . . . . 17

**2** *Ground Stations*

**Looking up from Earth** . . . . . **21**

2.1 Overview of Ground Stations . . . . . 22

2.2 SatNOGS Ground Stations . . . . . 22

2.3 SatNOGS Optical Ground Stations . . . . . 22

**3** *Toolchain and Pipeline*

**Make Old Photons Count** . . . . . **27**

3.1 SatNOGS Optical HOWTO . . . . . 28

3.2 Toolchain . . . . . 28

3.3 Hardware . . . . . 28

**4** *Hardware*

**Design and Set Up Equipment** . . . . . **33**

4.1 Overview of Hardware . . . . . 34

4.2 Camera . . . . . 34

4.3 Lenses . . . . . 35

4.4 Embedded System . . . . . 35

4.4.1 Embedded Systems Comparison . . . . . 35

4.5 Example Optical Ground Station with Tracking . . . . . 35

4.6 Tripods . . . 40
4.7 Mounts . . . 40
4.7.1 Sidereal Tracking Mounts . . . 43
4.7.2 Satellite Tracking Mounts . . . 43
4.8 Future Designs . . . 44

**5** ***Software***
**Satellite Applications on Earth . . . 45**
5.1 Software Overview . . . 46
5.2 Setup Optical Ground Station Software . . . 46
5.3 Configure Software . . . 47
5.4 skymap . . . 48
5.5 Tracking Software . . . 50
5.5.1 Telescope Tracking Software . . . 51
5.5.2 Antenna Tracking Software . . . 51
5.5.3 Camera Tracking Software . . . 51
5.6 GNSS with `gpsd` . . . 52
5.7 NTP . . . 53
5.8 KStars with Tracking Mount . . . 54
5.9 `stvid` with Docker . . . 60

**6** ***Acquire Data***
**Twinkle, Twinkle, Little Starlink . . . 61**
6.1 Overview of Operation . . . 62
6.2 Setup with `stvid` . . . 62
6.3 `acquire.py` Output . . . 62

**7** ***Plate Solvers***
**Place from Pixels . . . 65**
7.1 Plate Solver . . . 66
7.2 Source Extractor . . . 66
7.3 astrometry.net . . . 66
7.4 Star Catalogues . . . 66
7.5 World Coordinate System . . . 67
7.6 SIP . . . 67
7.7 Plate Solving with `stvid` . . . 67

**8** ***Detect***
**Find Satellites . . . 69**
8.1 Satellite Detection with stvid . . . 70
8.1.1 `process_new.py` Usage . . . 70
8.1.2 `process.py` Usage . . . 70
8.1.3 `process_new.py Example Output` . . . 70
8.2 AstroImageJ . . . 70

**9** ***Identify***
**More Than Just Old Rocket Bodies . . . 73**
9.1 Overview of Satellite Identification . . . 74

9.2 `stvid`'s `process.py` for Identification . . . 74
9.3 Identifying Satellites with `satid` . . . 74
9.4 Identification with stvid . . . 74
9.5 Review of Images . . . 74
9.5.1 Unidentified Satellites . . . 74
9.5.2 Airplanes . . . 81
9.5.3 Configuration and Setup Errors in Images . . . 81
9.5.4 Image Processed with Bad TLE . . . 81
9.5.5 Image Acquired During Slew . . . 81
9.5.6 Light Outside FoV Artifact . . . 81

**10** ***Upload & Share***
**Send Data to the SatNOGS Network** . . . **89**
10.1 Push Data Upstream . . . 90
10.2 File Extensions . . . 90
10.3 Data Files . . . 90
10.3.1 `.fits` FITS Files . . . 90
10.3.2 `.cal` CAL Files . . . 91
10.3.3 `.cat` CAT Files . . . 91
10.3.4 `.fits.png` FITS PNG Files . . . 92
10.3.5 `catalog.png` catalog PNG Files . . . 92
10.3.6 `unid.png` Unidentified PNG Files . . . 93
10.3.7 `.dat` Data Files . . . 93
10.3.8 `stvid .csv` Files . . . 94

**11** ***Support***
**Help!** . . . **97**
11.1 Support . . . 98

**12** ***Contact***
**Email, Chat, Forum** . . . **99**
12.1 Contact . . . 100

**List of Abbreviations** . . . **101**

**Acronyms** . . . **103**

**Glossary** . . . **111**

**Bibliography** . . . **121**

**Index** . . . **127**

# List of Figures


1 SatNOGS Network. . . . . xii
2 Libre Space Foundation Website. . . . . xiii
3 SatNOGS Website. . . . . xiv

1.1 SatNOGS DB website. . . . . 16
1.2 SatNOGS DB RamSat CubeSat web page. . . . . 16
1.3 SatNOGS DB RamSat CubeSat telemetry web page. . . . . 18
1.4 UPSat Website. . . . . 19
1.5 List of observations of the RamSat CubeSat, example. . . . . 19
1.6 Sample observation of RamSat CubeSat on SatNOGS website. . . . . 20

2.1 SatNOGS ground station with antennas. . . . . 23
2.2 Ground station viewed on SatNOGS website, example. . . . . 24
2.3 SatNOGS Optical ground station prototype. . . . . 25

3.1 SatNOGS Optical Process Overview . . . . . 29

4.1 SatNOGS-Optical ground station prototype. . . . . 36
4.2 Camera enclosure, front side, showing glass and lens. . . . . 37
4.3 Camera enclosure, left side, showing hinge. . . . . 37
4.4 Camera enclosure, right side. . . . . 38
4.5 Camera enclosure, right side, opened. . . . . 39
4.6 Camera enclosure, opened. . . . . 41
4.7 Camera enclosure, top. . . . . 42

5.1 SatNOGS Optical Operation Pipeline . . . . . 46
5.2 `skymap` example screenshot. . . . . 49
5.3 KStars sky chart, example screenshot. . . . . 57
5.4 KStars using V4L2, example screenshot. . . . . 58
5.5 KStars using simulated CCD, example screenshot. . . . . 59

6.1 Satellite images acquired by stvid. . . . . 63

7.1 nova.astrometry.net website. . . . . 67

8.1 AstroImageJ screenshot viewing FITS file. . . . . 71

9.1 `satid` example. . . . . 75
9.2 Example satellite image processed by `stvid`. . . . . 76
9.3 stvid processed image, with three unidentified satellites. . . . . 77
9.4 stvid processed image, with three unidentified satellites, the first labeled with NORAD ID `90000`. . . . . 78
9.5 stvid processed image, with three unidentified satellites, the second labeled with NORAD ID `90001`. . . . . 79
9.6 stvid processed image, with three unidentified satellites, the third labeled with NORAD ID `90002`. . . . . 80
9.7 Airplane, bold left side, in stvid processed image. . . . . 82
9.8 Airplane, blinking, in stvid processed image. . . . . 83
9.9 Airplane, two solid lights plus blinking, in stvid processed image. . . . . 84
9.10 `stvid` processed image, showing mis-configured/setup system. . . . . 85
9.11 `stvid` processed image, alignment off due to use of old TLE. . . . . 86
9.12 `stvid` processed image, trails due to slewing during acquire. . . . . 87
9.13 `stvid` processed image, with classified satellites and light artifact. . . . . 88

10.1 `stvid` PNG. . . . . 92
10.2 `stvid` PNG of the satellite trail of NORAD ID 48473 identified in red. . . . . 93
10.3 `stvid` PNG of the satellite trail of NORAD ID 52718 identified in red. . . . . 94
10.4 `stvid` PNG of an unidentified trail in red. . . . . 95

# List of Tables

4.1 Comparison of embedded systems . . . . . 35

# Introduction

# SatNOGS Optical

The SatNOGS Optical Unofficial Guide documents how to set up and run an optical ground station for the LSF's SatNOGS network.

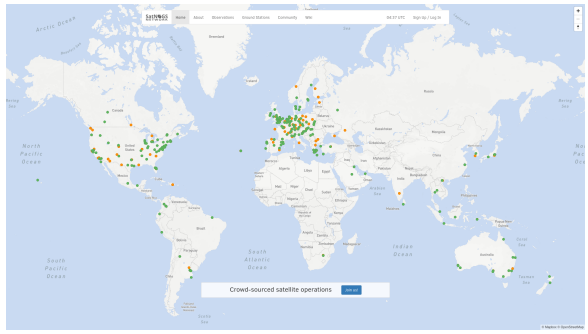



**Figure 1:** SatNOGS Network.

## Unofficial Guide Overview

First, an overview of the this text, then the LSF and some of it's key projects, such as the SatNOGS network. A cursory review of satellites and existing RF ground stations will be followed by a big picture view of acquiring and processing images of satellites. Hardware is reviewed, then software to run on it, with many options, including what is best. Finally, what to do with the data. The chapters that follow are listed below.

**Satellites** — What are we looking at?

**Ground stations** — How Earth talks to satellites and back.

**Toolchain and Process** — The big picture of what hardware and software is needed to set up an optical ground station for use on the network.

**Hardware** — Details on appropriate hardware configurations, and example setups.

**Software** — A look at the myriad software related to satellites, and what works best at present for SatNOGS Optical.

**Acquire** — Convert photons to bits. Pointing a camera at the sky works.

**Solve** — Pictures of stars reveal the time and location of the photo. Plate solvers reviewed.

**Detect** — The plate solver says where the photo is, now detect if are there moving trails that aren't star trails that could be satellites.

**Identify** — With time, location, satellite detection, TLEs are overlaid and compared with detected satellites. Satellite identification by computers and humans.

**Upload** — When ready, data will be pushed to the SatNOGS network.

**Support** — Where development is occurring and questions answered!

## Libre Space Foundation

The LSF supports "free and accessible space for all, creating Open Source space technologies."[1]

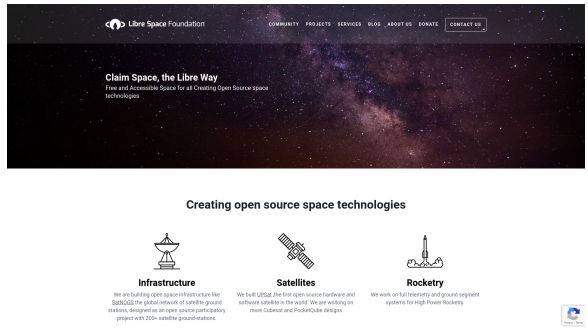



**Figure 2:** Libre Space Foundation Website.

Select LSF projects:

**SatNOGS** — Global network of satellite ground stations
`https://satnogs.org`

**UPSat** — First OSH and OSS satellite in the world
`https://upsat.gr/`

## SatNOGS Network

SatNOGS is the LSF's global network of satellite ground stations.

## Optical Network Status

The SatNOGS RF network has been running successfully for years. Adding a SatNOGS Optical network is a new development. At present there are zero nodes on the network.

[1] `https://libre.space`

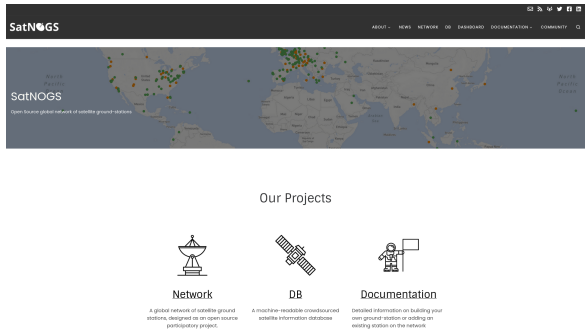



**Figure 3:** SatNOGS Website.

Some software already exists, some is being ported from C to Python, and other parts remain to be done. There is software available for acquiring optical data of satellites. There is no facility at present for pushing data back to the network.

# Satellites

# Artificial Satellites in Space

## 1.1 Overview of Satellites

This chapter gives a brief overview of satellites, with particular antention to ones using amateur radio bands.

## 1.2 SatNOGS DB

The SatNOGS DB is a database that lists satellites, many of which can be tracked by ground station operators on the SatNOGS network.

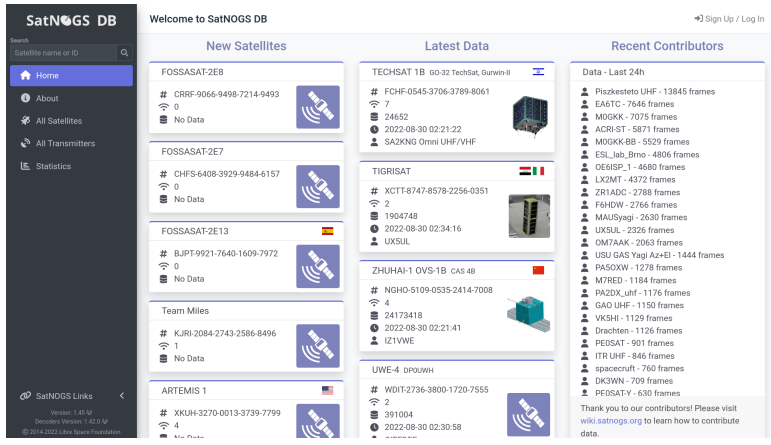



**Figure 1.1:** SatNOGS DB website.

Individual satellites can be viewed, such as the RamSat CubeSat.

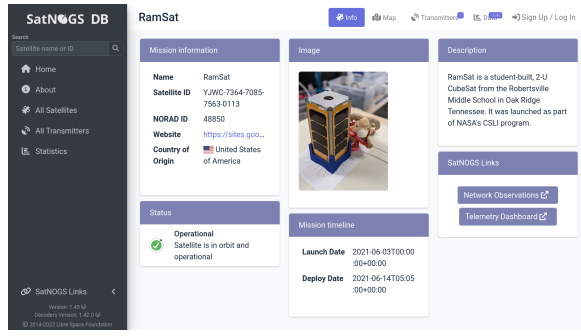



**Figure 1.2:** SatNOGS DB RamSat CubeSat web page.

For some satellites, telemetry is available in a Grafana dashboard. See the RamSat telemetry dashboard.

## 1.3 Libre Satellites

Below are satellites in orbit, decayed, or designs on Earth that generally support the libre way. These are some select libre satellites:

**UPSat** — First libre OSH and OSS satellite in the world. Created by the LSF.
`https://upsat.gr/`

RF Observations can be scheduled on the SatNOGS network. See figure 1.5, page 19 for a example list of observations of the RamSat CubeSat.[1]

Individual RF observations are uploaded to the SatNOGS network, as can be seen in the example observation of the RamSat CubeSat by SatNOGS ground station "2380 - Piszkesteto UHF" run by volunteer bcsak (username).

## 1.4 Satellite Orbits

Satellite orbits. See SGP and SDP.

See TLE.

[1] `https://network.satnogs.org/observations/?future=0&norad=48850&observer=&station=&start=&end=&transmitter_mode=&page=2`

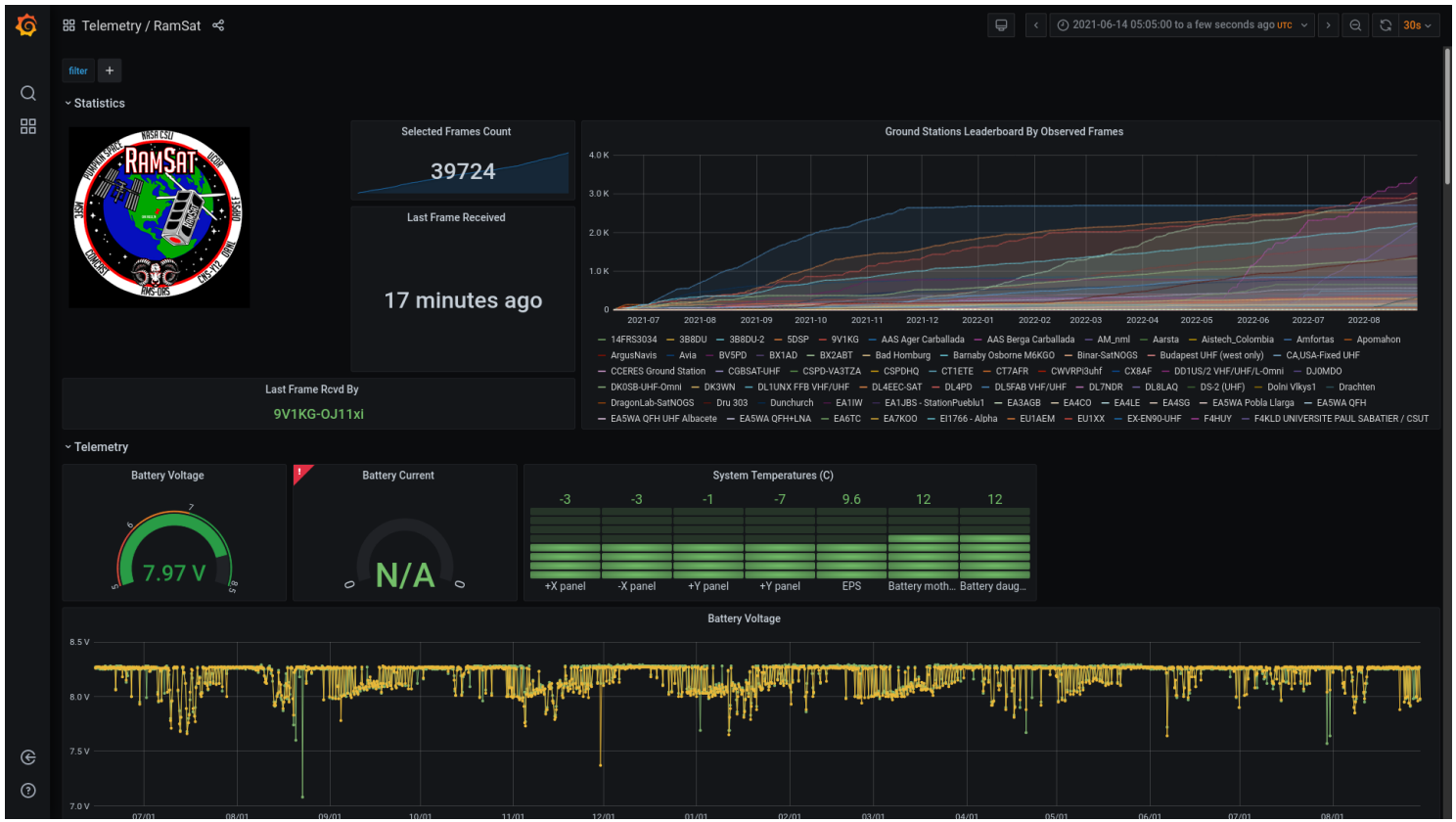



**Figure 1.3:** SatNOGS DB RamSat CubeSat telemetry web page.

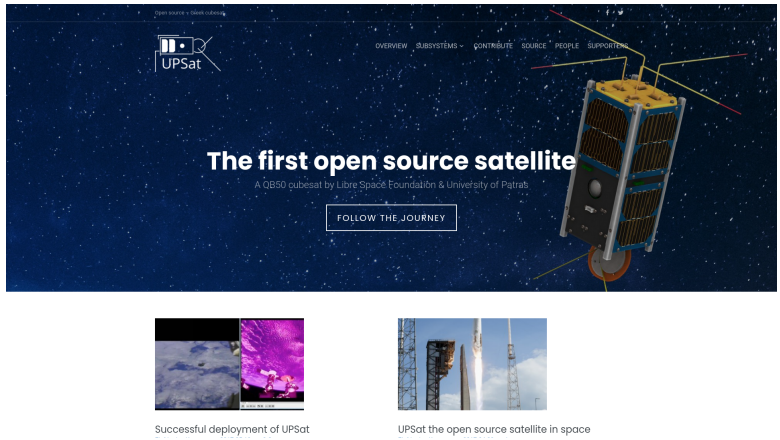



**Figure 1.4:** UPSat Website.

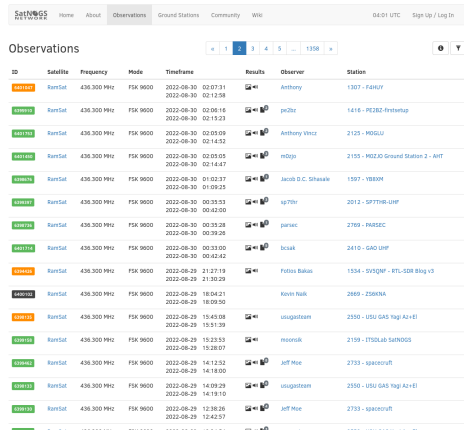



**Figure 1.5:** List of observations of the RamSat CubeSat, example.

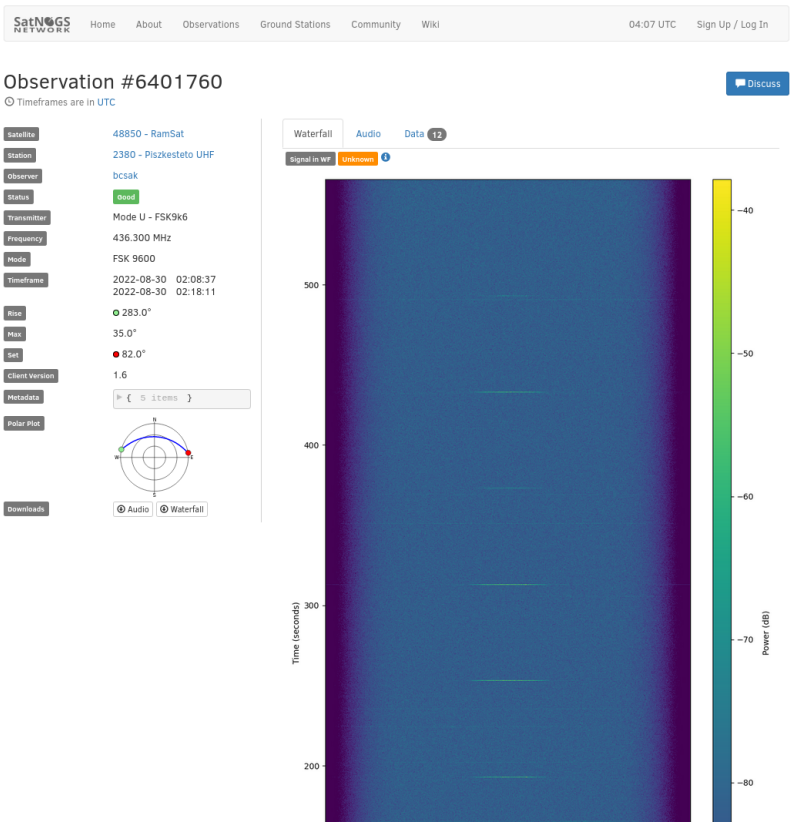



**Figure 1.6:** Sample observation of RamSat CubeSat on SatNOGS website.

# Ground Stations

# Looking up from Earth

## 2.1 Overview of Ground Stations

Ground stations are a setup of equipment such as embedded systems, cameras, SDRs, antennas, and receivers, located on Earth, observing space.

## 2.2 SatNOGS Ground Stations

The LSF built the SatNOGS network using distributed, volunteer-run, RF ground stations using SDRs, such as in figure 2.1, page 23.[1] It shows a SatNOGS ground station with VHF (right) and UHF (left) antennas on a mast with an altazimuth mount rotator. SDRs can be used as RF receivers.[2]

Ground stations can be viewed on the SatNOGS network website, such as as the example in figure 2.2, page 24.[3]

## 2.3 SatNOGS Optical Ground Stations

The LSF is developing SatNOGS Optical to add optical ground stations to the network.

Prototype optical ground stations are being developed. An example setup, using a Sky-Watcher EQ6-R Pro telescope tripod and tracking mount, can be seen in figure 2.3, page 25.

[1] https://spacecruft.org/assets/i/spacecruft-rotator.png

[2] Wikipedia contributors. *Software-defined radio — Wikipedia, The Free Encyclopedia.* https://en.wikipedia.org/w/index.php?title=Software-defined_radio&oldid=1107093398. [Online; accessed 28-August-2022]. 2022

[3] https://network.satnogs.org/stations/2733/

**Figure 2.1:** SatNOGS ground station with antennas.

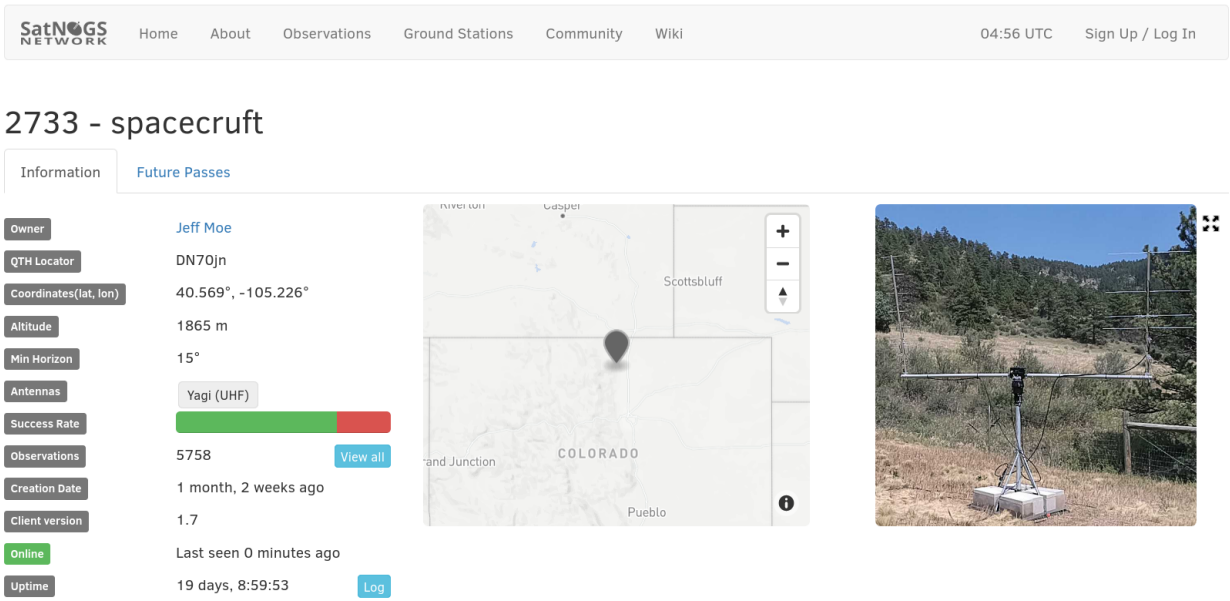



**Figure 2.2:** Ground station viewed on SatNOGS website, example.

**Figure 2.3:** SatNOGS Optical ground station prototype.

# Toolchain and Pipeline
# Make Old Photons Count

## 3.1 SatNOGS Optical HOWTO

SatNOGS Optical is the nascent distributed network of optical ground stations.

This chapter gives a top level review what is needed in terms of hardware and software to build an operating optical ground station.

## 3.2 Toolchain

See below for a SatNOGS Optical Process Overview.[1] See figure 3.1, page 29, described below.

**Hardware** — Hardware, such as cameras and embedded systems, is to be selected and set up.

**Software** — The best currently available software is to be downloaded, installed, and configured.

**Acquire** — Data samples, typically in the form of FITS file photographs, need to be acquired by running a camera outside at night taking pictures of the sky.

**Plate solver** — Acquired data samples in FITS files need to be processed by a plate solver. See section 7.1, page 66.

**Detect satellites** — Using TLEs and the "solved" plates, detect satellites. See section 8.1, page 70.

**Identify satellites** — With satellites detected in the previous step, identify what they are. See 9.1, page 74.

## 3.3 Hardware

Discussed in this section are some of the hardware options to be explored. More explicit instructions of a particular hardware installation can be see in section 4.1, page 34. Discussed below are camera options, for details on embedded systems and other parts, also see hardware in section 4.1, page 34

For the purposes here, there are three main categories of hardware. Depending which category of equipment is selected, it impacts everything else, such as the software used. Main categories:

**Motion video cameras** — Moving images.

**Still camera** — Still photos.

**Allsky cameras** — Views of all, or nearly all of the sky.

[1] `https://spacecruft.org/spacecruft/SNOPO`

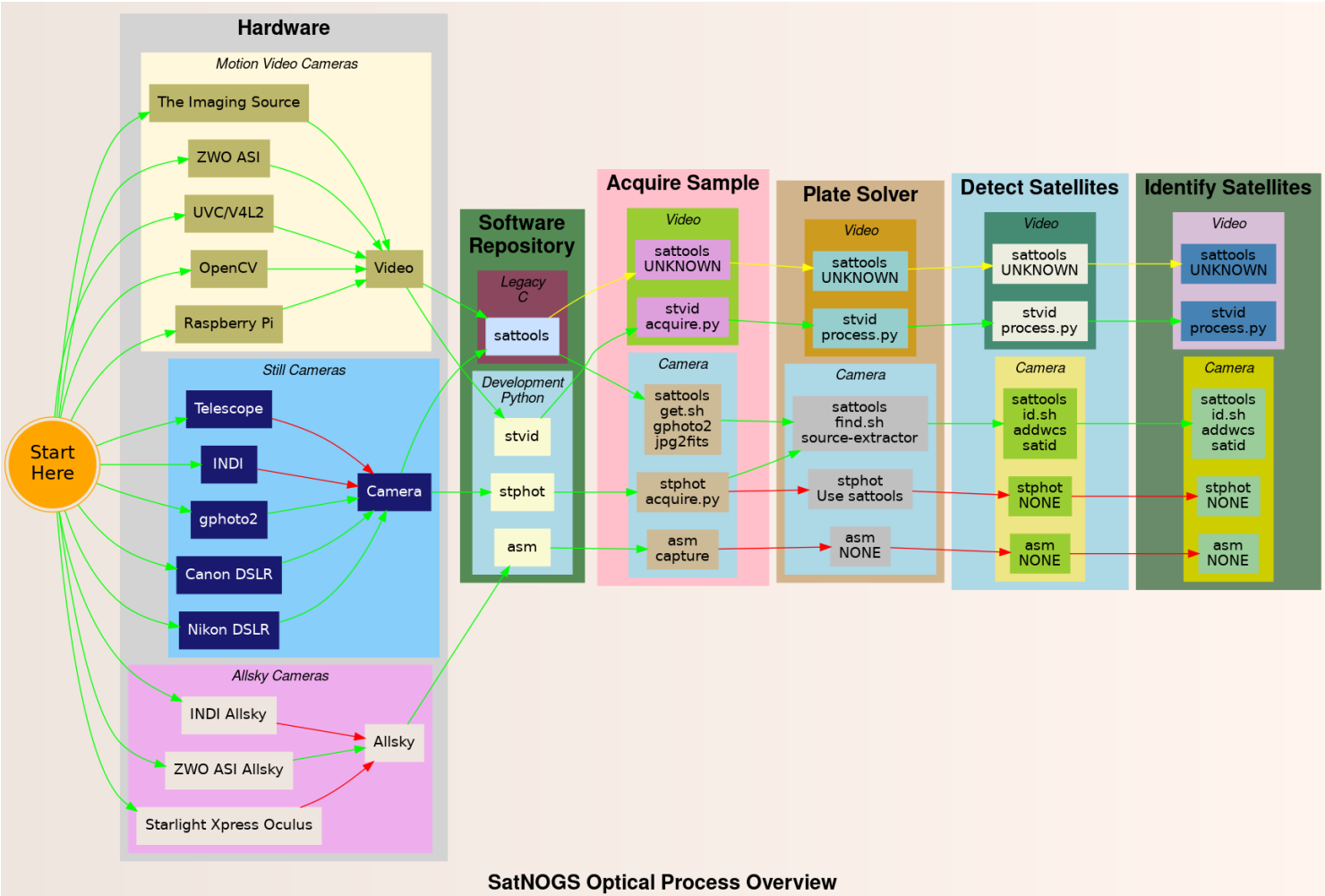



**Figure 3.1:** SatNOGS Optical Process Overview

Different types of equipment can be used in different categories. Some can be used in multiple setups, most just in one. If available, using motion video cameras will work best for detecting satellites with the developing SatNOGS toolchain. Examples of motion video camera sources that could be used:

**The Imaging Source Cameras based on IMX174** — Known to work. Recommended. High quality cameras, believed to be usable following DFSG.

**ZWO ASI based on IMX174** — Known to work. Not DFSG compatible. Uses proprietary SDK. Currently in prototype development.

**UVC/V4L2** — "Any" video camera that works with the Linux kernel. Typically, the device will appear similar to `/dev/video0`. A camera that works with the software isn't necessarily sensitive enough to detect satellites, however, as most are designed for brighter environments.

**OpenCV** — cameras that work with OpenCV can be used, same as UVC. To work well, they need to be sensitive.

**Raspberry Pi** — The PiCamera can be used. A good lower cost option. Recommended. Many non-Raspberry Pi devices, such as Odroid are also compatible with the Pi MIPI interface.

Still cameras can also be used productively. The current Python toolchain is in very early development and not completely usable yet.

See the list below for still camera options:

**Telescopes** — Can definitely take images of satellites. Not the best tool at present, as it isn't well integrated into the SatNOGS Optical toolchain. The FoV is generally too small. The mounts are optimized for different types of tracking than satellites. This is changing, and longer term could be well-supported. Using RASA style astrographs is likely the best option.

**INDI** — Typically used for control of telescopes and associated instrumentation, such as tracking mounts and cameras. Cannot be used directly with the current developing SatNOGS toolchain. It is not seen as the future path forward as it isn't well optimized for SatNOGS Optical usage. That said, it is very useful at present for running a tracking mount with KStars and Ekos, for example, in lieu of a better option. Camera software in the INDI platform typically produce image FITS files.

**gPhoto** — The Linux kernel recognizes many cameras that can be used with gPhoto tools and drivers, available in Debian. This is the recommended option at present for still cameras. DSLR cameras, such as from major manufacturers Canon and Nikon, are used with gPhoto.

Considering the hardware options above, they need to be matched with corresponding software. Not all options work (at all), and some cannot be easily used to perform all steps needed.

There are also broader "paths" that need to be considered:

**sattools** — Deprecated because it is in C, and the decision by upstream and the LSF was made to move forward with applications primarily written in Python. Sattools is the most complete toolkit, however, so no matter what path is chosen, some parts of it will likely be used for now. It can be used with motion video cameras and still cameras. It includes many other software tools related to satellites.

**stvid** — This is the best path if a motion video camera is available. It is in Python and is the tool the SatNOGS Optical project is using as the basis for future development. It still depends on some C tools from sattools.

`stphot` — Written in Python this is what the SatNOGS Optical project will likely use in the future for still cameras. It is in very early development, but can acquire data (take photos) with gPhoto-compatible cameras.

`asm` — All Sky Monitor for taking pictures of all, or nearly all of the sky, such as with a 150 or 180 degree view. The `asm` application is in pre-development, but is in Python and could be the basis for future SatNOGS Optical development. The difficulty with all sky cameras is the plate solver isn't written to use images from the "fish-eye" view of an all sky camera.

**Other** — There are many other satellite and telescope software packges freely available on the Internet. Perhaps some could be adapted for usage.

# Hardware

# Design and Set Up Equipment

## 4.1 Overview of Hardware

Hardware considerations for a SatNOGS Optical ground station.

Main hardware components in an optical ground station:

- Lens.
- Camera.
- Embedded system (computer).

Other components:

- Ethernet cable.
- USB cable.
- Enclosure.
- Power supply.
- Tripod.
- Manual or tracking mount.
- Power source, grid or alternative.
- Internet, wifi or ethernet.

## 4.2 Camera

Cameras being evaluated:

- The Imaging Source IMX174 based, monochrome.
- ZWO ASI174MM, monochrome.
- ZWO ASI1600MM Pro, monochrome.
- DSLR camera.
- PiCamera.

NOTICE:
ZWO/ASI cameras require proprietary non-libre software on host computer and is not DFSG compatible.

## 4.3 Lenses

For lenses, the faster the better. F1.2 works well. F1.8 is the maximum recommended.

Lenses being tested:

- Kowa 50mm f1.4 C-mount.
- Canon EF 50mm f1.2 USM.

## 4.4 Embedded System

Embedded systems, such as Raspberry Pi, that can be used.

**Odroid N2** — Confirmed working.

**Odroid M1** — Testing.

**Raspberry Pi 3** — ?

**Raspberry Pi 4** — ?

**Intel NUC** — ?

### 4.4.1 Embedded Systems Comparison

Comparing embedded systems for SatNOGS Optical.

| Make | Model | Architecture | Max RAM | eMMC | |
|---|---|---|---|---|---|
| Odroid | N2 | ARM64 | 4 GB | Yes | |
| Odroid | M1 | ARM64 | 8 GB | Yes | |

**Table 4.1:** Comparison of embedded systems

## 4.5 Example Optical Ground Station with Tracking

This is an example of a tracking ground station. It is a prototype, so there are lots of mis-matched, overbuilt/underbuilt parts. The full setup on tripod, can be seen in 2.3, page 25.

A close up of the setup can be seen at 4.1, page 36, showing the Sky-Watcher telescope tracking mount, a Bosch PoE camera enclosure, and through the glass the camera lens. In the background is a white antenna for GNSS (GPS) and a solar power setup.

The camera lens is protected by the enclosure glass, which is IP67 (XXX) rated. See a close up of the front of the enclosure and camera lens in figure 4.2, page 37.

**Figure 4.1:** SatNOGS-Optical ground station prototype.

**Figure 4.2:** Camera enclosure, front side, showing glass and lens.

**Figure 4.3:** Camera enclosure, left side, showing hinge.

As seen in figure 4.3, page 37, the left side of the enclosure has a hinge for opening. The bottom white component is part of the telescope mount.

Figure 4.4, page 38, shows the right side of the enclosure. Two mounting bolt access points can be seen on each end. These are unscrewed with a hex head tool (supplied) to open the enclosure.

**Figure 4.4:** Camera enclosure, right side.

The enclosure is opened from the right side, as shown in figure 4.4, page 38.
Inside the camera enclosure, as shown in Figure 4.6, page 41, is:

- The Imaging Source DMX camera with Sony IMX174 CMOS.
- Kowa 50mm f1.4 C-mount lens.
- Odroid N2 running Debian GNU Linux system.
- Odroid N2 plastic enclosure, large half, hole drilled for ad-hoc mounting.
- Blower fan on top, with power cable (came with Bosch enclosure).
- Fan, maybe not so useful, with power cable (came with Bosch enclosure).
- Camera mounting plate (came with Bosch enclosure).
- Camera mounting screws, M6x25 (?).
- Ethernet cable, internal, short white (came with Bosch enclosure).
- PoE ethernet cable, external, plugged into PoE switch for data and power.
- USB 3 cable, internal, way too long, needs replacing, from Odroid to camera. XXX flat connector
- USB 3 cable, external, from Odroid to telescope mount. XXX large rectangle connector
- "Custom" 12Volt (V) DC power cable from Bosch PoE to Odroid.

**Figure 4.5:** Camera enclosure, right side, opened.

- Assorted nuts, bolts, and washers for an ad-hoc standoff height.

The top of the enclosure shows weather protection and a sun visor. See figure 4.7, page 42.

## 4.6 Tripods

The camera setup can be mounted a wide variety of ways, from just setting the camera somewhere (worst option), to a heavy duty pier with tracking mount (best option).

At present, most prototype optical ground stations are using static mounts on tripods.

Tripod and similar options include:

**No mount** — Quick and dirty, just hang the camera out somewhere sitting on something.

**Small tripod** — There are small desk tripods than can be used with lighter setups, such as used with a Raspberry Pi PiCamera.

**Photography Tripod** — Using a common camera tripod, of which there is a wide variety, from light to heavy.

**Telescope Tripod** — Similar to photography tripods, but typically heavier weight.

**Telescope Portable Pier** — Similar to a telescope tripod, but much heavier, typically with a larger center pier post. Still movable, and folds up similar to a photography tripod.

**Telescope Pier** — A wide variety, such as making a roughly 1.5 meter permanent cement post.

## 4.7 Mounts

For mounts, there are two main types: tracking or static. By the latter "static" mounts, it is meant that the camera, the tripod, and the mount all stay motionless. This is what you would get using a camera with a common photography tripod and a simple mounting plate.

Static mounting options include:

**Camera plate** — Commonly used on cameras and readily available for mounting cameras to tripods.

**Enclosure plate** — A flat plate with holes drilled in it to mount the camera inside an enclosure.

**"Security" camera enclosure mount** — Various mounts exist to mount security cameras to posts, walls, etc.

Tracking mount options to consider include:

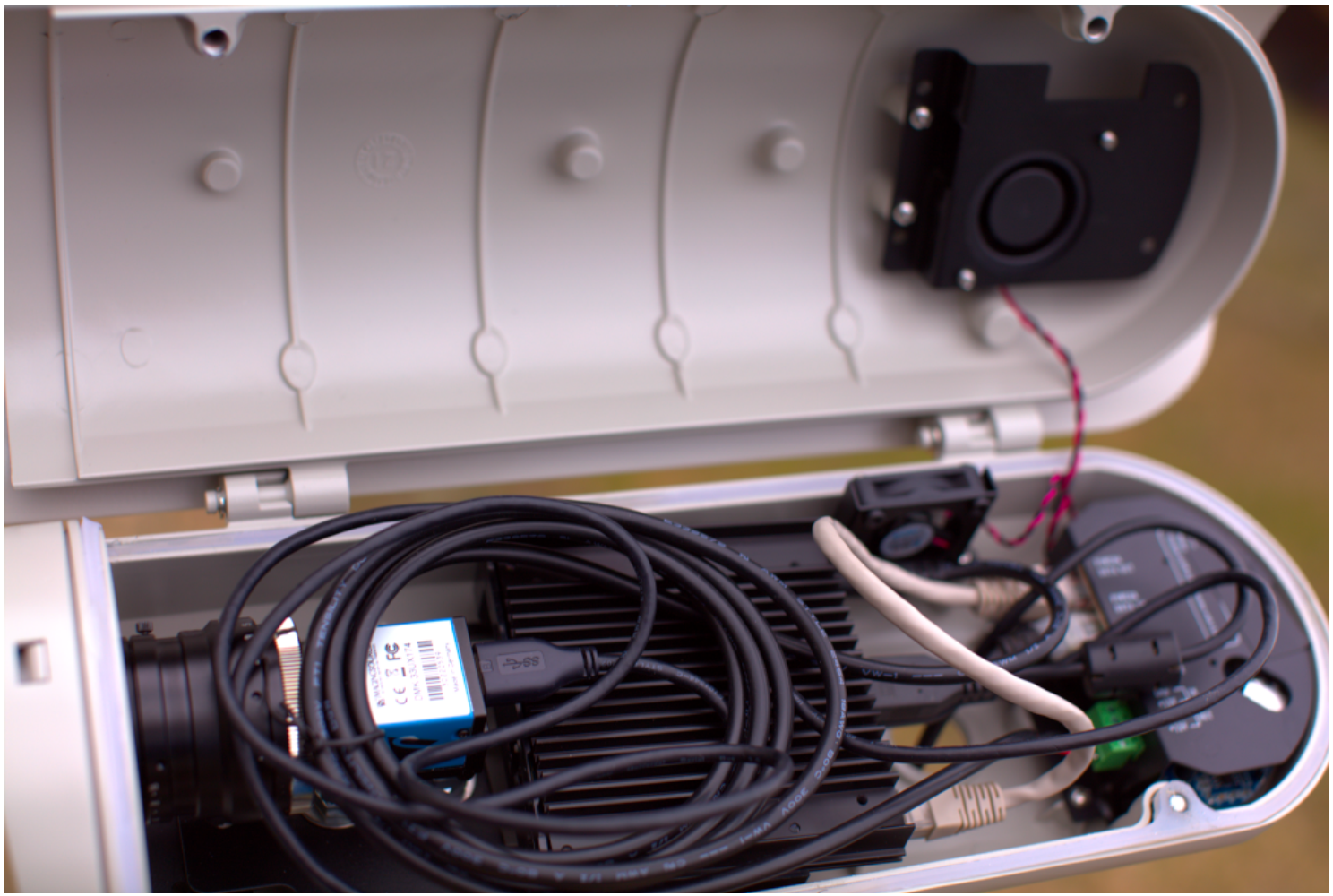


**Figure 4.6:** Camera enclosure, opened.

**Figure 4.7:** Camera enclosure, top.

**Sky-Watcher EQ6-R Pro** — Telescope mount using INDI.

**Celestron** — Wide variety of telescope mounts using INDI.

**iOptron** — Telescope mount with (untested) satellite tracking.

**INDI Telescope Mounts** — A wide variety of other INDI compatible telescope mounts.

**Yaesu G-5500** — Antenna rotator.

**hamlib** — Other hamlib compatible rotators.

**Teledyne FLIR PTU-5** — High Performance PTZ Unit designed for security cameras (untested, no drivers?).

**Misc PTZ** — Other security camera PTZ mounts.

Tracking mounts aren't widely used, but there is support for them in `stvid` when acquiring data. The tracking needs to be set up independently of `stvid`. At present, I use KStars with Ekos to control a Sky-Watcher EQ6-R Pro tracking mount.

For tracking, there a few different ways to track:

**Static** — No tracking, just point at one place in the sky. Generates star trails. Generates satellite trails.

**Sidereal tracking** — Tracks stars. Generates stars as points. Generates satellite trails.

**Satellite tracking** — Tracks satellites. Generates star trails. Generates satellites as points or potentially larger images of the satellite structure.

### 4.7.1 Sidereal Tracking Mounts

Sidereal tracking ("telescope tracking") is what COTS tracking "GoTo" telescopes from Celestron or Sky-Watcher do, for example. They track the stars, countering the rotation of the Earth to keep the same view of the sky in the camera's FoV. Stars remain as points, even after multi-minute or multi-hour imaging. This is what is used for "pretty" pictures of stars, nebula, galaxies, etc. This is the most common tracking set up, as it has been widely used in astronomy communities for decades.

Within sidereal tracking mounts, there are yet more options:

**Fork** — Fork mount.

**EQ fork** — Fork on EQ mount.

**German EQ** — Most common EQ mount.

**More** — Endless variety of available telescope mounts.

Also related to sidereal tracking is lunar and planetary tracking, but for our uses all three will be included under sidereal tracking.

To use a sidereal tracking mount for imaging satellites, the camera must "leap frog" the satellite. At present, my practice is to use a sidereal mount, point at a location with KStars, start stvid. Then stop stvid, move to new location using KStars, start stvid.

See Software section 5.5, 50 for information on using tracking mounts.

### 4.7.2 Satellite Tracking Mounts

Of the options between a static mount (no tracking), sidereal tracking, and satellite tracking, the latter is by far the least common. In this case, the tracking mount is tracking the satellite itself. This is much more complex than tracking stars, which it builds upon. It requires, such as:

**Time** — Accurate time, such as from GNSS or NTP.

**Location** — Accurate location, also available from GNSS.

**TLE** — Need to know the satellites' orbit (accurately!).

**Variable speed tracking** — Satellites are moving at different speeds above, the mount needs to be capable of that.

**Alignment** — On top of all the gear and software needed, the equipment needs to be accurately aligned.

Most tracking equipment for telescopes, cameras, and antennas usually has just a few speeds, such as a slew speed and a sidereal star tracking speed. Sometimes there will be a few steps of these speeds (e.g. slew speeds from 1-9), but not the finely tuned tracking speeds needed to track a satellite. Oftentimes the telescope tracking maximum speed will be too slow for satellite tracking.

Variable speed tracking (XXX phrase?) is needed for tracking satellites if the goal is to keep the satellite in the (near) center of the image frame and leave star trails. The speed the mount moves needs to be calculated based upon a recent orbit calcuation, such as from a TLE.

There are highly skilled amateur astronomers that have captured detailed pictures of artificial satellites, such as the ISS and astronauts doing space walks, using hand guided telescopes with low cost CCD imagers.

There are few options for satellite tracking mounts. Some new iOptron telescope mount firmware supports tracking satellites. This has been largely untested so far, but at present is likely the best option, if a satellite tracking mount is wanted.

## 4.8 Future Designs

There is some discussion of using much larger "lenses", such as a RASA "telescope" (See: astrograph). The primary concern is the lack of satellite tracking mounts, because telescope mounts are generally too slow, and need to "leap frog" the satellite. Rotators used for antennas aren't typically stable enough for a camera.

# Software

# Satellite Applications on Earth

## 5.1 Software Overview

See 5.1, page 46 for a diagram with an overview of the software process.

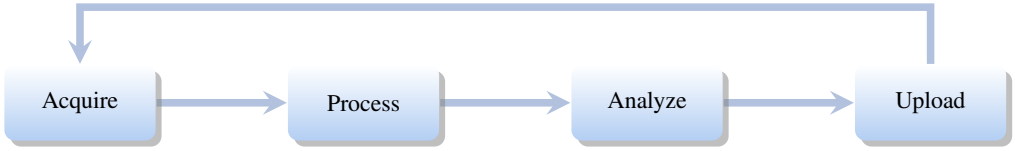



**Figure 5.1:** SatNOGS Optical Operation Pipeline

## 5.2 Setup Optical Ground Station Software

HOWTO set up and configure a SatNOGS Optical ground station prototype. Note: the SatNOGS network is not yet ready for optical data.

For a description of installation using Docker, see section 5.9, page 60. For a "manual" install, see immediately below.

Setup an embedded system, such as a Raspberry Pi or an Odroid N2, with Debian stable (11/Bullseye) or testing (Bookworm).

See each software repository for latest documentation.

Install dependencies from the Debian software repository:

```
sudo apt update
sudo apt install python3-virtualenv python3-pip python3-dev \
    source-extractor astrometry.net \
    libeigen3-dev giza-dev libx11-dev \
    astrometry-data-tycho2 astrometry-data-tycho2-07 \
    astrometry-data-tycho2-08 astrometry-data-tycho2-09 \
    astrometry-data-tycho2-10-19
```

Next, install the `hough3d-code` dependency.

```
git clone https://gitlab.com/pierros/hough3d-code
cd hough3d-code
make all
sudo cp -p hough3dlines /usr/local/bin/hough3dlines
```

Install `satpredict` from using either the cbassa or spacecruft software repository.

```
cd ../
git clone https://spacecruft.org/spacecruft/satpredict
# Or
#git clone https://github.com/cbassa/satpredict
cd satpredict/
make
sudo make install
```

Now install `stvid`, the main acquisition and processing application. It is written in Python. Either use the spacecruft `git` software repository or the cbassa one.

```
cd ../
git clone https://spacecruft.org/spacecruft/stvid
# Or use upstream:
#git clone https://github.com/cbassa/stvid
```

Optionally, set up a Python virtual environment and use PIP:

```
cd stvid
virtualenv -p python3 env
source env/bin/activate
pip install --upgrade pip
```

Install dependencies. Numpy needs to be installed first, or the install fails.

```
pip install --upgrade numpy
pip install --upgrade -r requirements.txt
```

If the system `python3-numpy` conflicts, you could try installing numpy with PIP in the virtualenv thusly:

```
pip install --upgrade --ignore-installed numpy
```

Debian in bug `#941466` changed the name of the `sextractor` dependency to `source-extractor`, so a symlink is needed for now:

```
sudo ln -s /usr/bin/source-extractor /usr/local/bin/sextractor
```

## 5.3 Configure Software

Configure the embedded system.

```
cd stvid/
cp -p configuration.ini-dist configuration.ini
# Edit
# Note for fields st-username and st-password, *don't* use quotes.
# Set:
vi configuration.ini
```

Edit various `configuration.ini` fields:

```
# For example:
observer_cospar = 9999          # COSPAR number of observer's site
observer_name = Jeff Moe        # Name of observer
observer_lat = 40.568912        # Latitude of location in decimal degrees
observer_lon = -105.225852      # Longitude of location in decimal degrees
observer_height = 1860           # Elevation of location in meters

observations_path = ./data/obs # Path for raw acquisitions
tle_path = ./data/tle           # Path where TLEs are stored (and updated)
```

```
results_path = ./data/results    # Path for results of processed obs

st-username = foo
st-password = bar

# Path to source-extractor
sex_config = /usr/share/source-extractor/default.sex
```

Store downloads here:

```
mkdir -p ./data/obs ./data/tle ./data/results
```

The `ST_DATADIR` variable is still used, from `sattools`. Maybe set all these in `~/.bashrc` (suit to fit directory setup):

```
export ST_DATADIR=$HOME/sattools
export ST_TLEDIR=$HOME/TLE
export ST_OBSDIR=$HOME/satobs
export ST_LOGIN="identity=foo@no:pass"
```

Set `astrometry.net` to run in parallel, assuming you have enough RAM: (This doesn't appear to work? Breaks?).

```
sudo vim /etc/astrometry.cfg
# Uncomment:
inparallel
```

## 5.4 skymap

Use skymap for viewing satellite orbits trails projected on a map of of the sky. Skymap is part of sattools. Skymap isn't a required part of the toolchain, but it is useful to see what satellites are visible at a particular time and location.

Source:

* `https://github.com/cbassa/sattools`

My fork:

* `https://spacecruft.org/spacecruft/sattools`

> NOTICE:
> The main upstream sattools package requires non-libre `pgplot` on the host computer which is not DFSG compatible.

Described below is how to build and install my fork of sattools to use skymap instead of upstream. Upstream requires non-libre software which also happens to be very difficult for most users to compile. My fork uses the Giza library which replaces pgplot. It is much easier to build.

```
# Install dependencies
sudo apt install giza-dev git make dos2unix source-extractor wcslib-dev \
     libgsl-dev gfortran libpng-dev libx11-dev libjpeg-dev libexif-dev

# Clone repo
git clone https://spacecruft.org/spacecruft/sattools
```

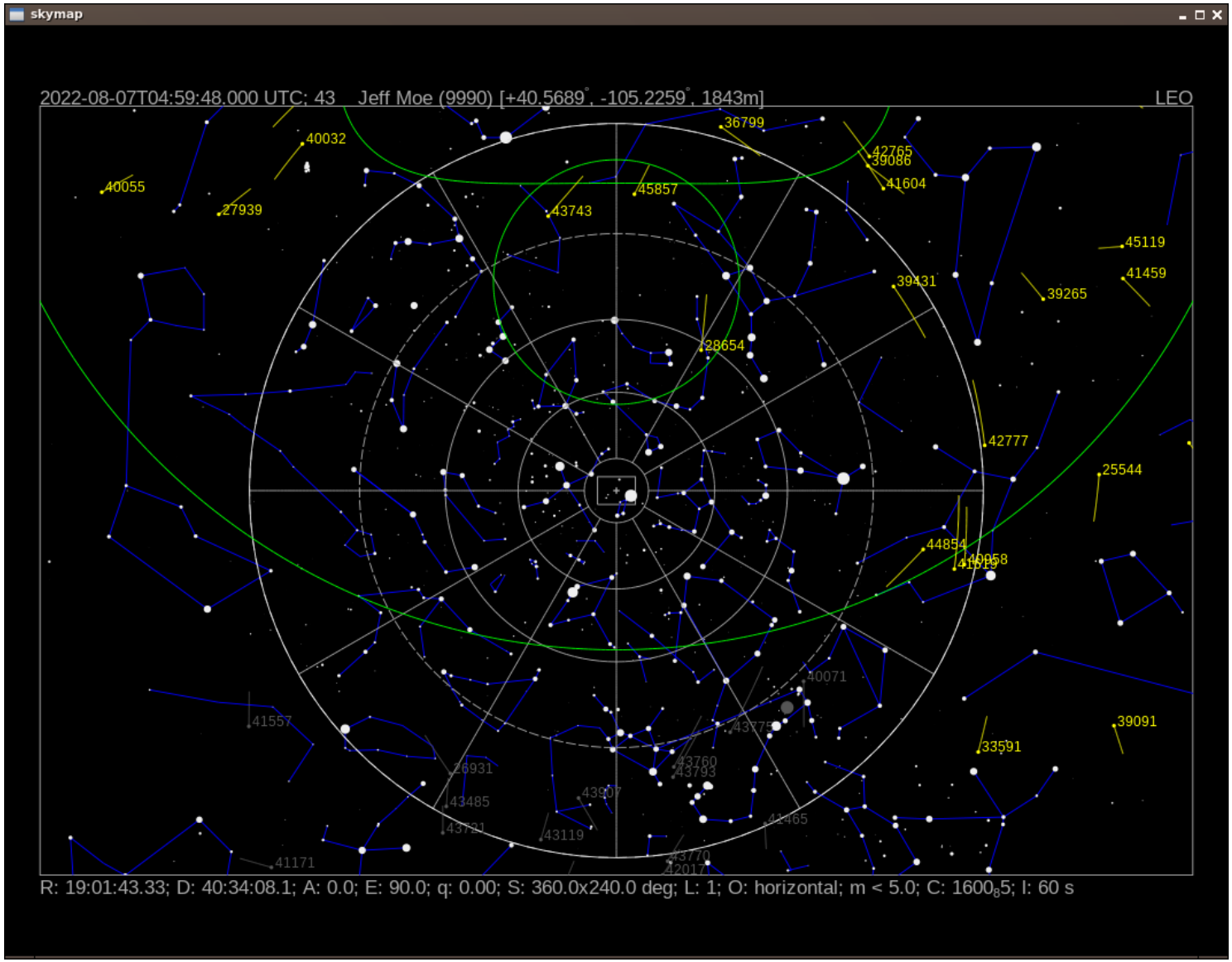



**Figure 5.2:** `skymap` example screenshot.

```
# Or use upstream (needs more steps)
git clone https://gitub.com/cbassa/sattools

cd sattools/
# Build
make

# Install
sudo make install

# Clean
make clean

# To rebuild
make clean
make

# To uninstall
sudo make uninstall
```

See below for skymap (fork) usage:

```
tleupdate

# set config in ~/.bashrc XXX

skymap --help
Usage: skymap [OPTION]
Visualize satellites on a map of the sky.

-t, --time    Date/time (yyyy-mm-ddThh:mm:ss.sss) [default: now]
-c, --catalog    TLE catalog file [default: satnogs.tle]
-i, --id    Satellite ID (NORAD) [default: all]
-R, --ra    R.A. [hh:mm:ss.sss]
-D, --decl    Decl. [+dd:mm:ss.ss]
-A, --azimuth    Azimuth (deg)
-E, --elevation    Elevation (deg)
-w, --width    Screen width (default: 1024). Set height too.
-g, --height    Screen height (default: 768). Set width too.
-n, --nmax    nmax line resolution/speed (default 128)
-S, --all-night    All night
-Q, --no-stars    No stars
-a, --all-objects    Show all objects from catalog (default: LEO)
-h, --help    This help
-s, --site    Site (COSPAR)
-d, --iod    IOD observations
-l, --length    Trail length [default: 60s]
-P, --planar-id    planar search satellite ID
-r, --planar-alt    planar search altitude
-V, --visibility-alt    altitude for visibility contours
-p, --positions-file    File with xyz positions
-L, --longitude    manual site longitude (deg)
-B, --latitude    manual site latitude (deg)
-H, --elevation    manual site elevation (m)
```

## 5.5 Tracking Software

The tracking software used depends upon the tracking mount.

See section 4.5, page 35 for discussion of hardware tracking mounts.

For our uses here, there are three broad categories of tracking software:

**Telescopes** — Controlling telescopes remotely.

**Antennas** — Controlling antennas remotely with hamlib.

**Cameras** — Controlling PTZ cameras remotely.

### 5.5.1 Telescope Tracking Software

Software that can be used with telescope tracking mounts:

**INDI** — Main client/server used by other applications.

**KStars** — Sky charts, INDI control.

**Ekos** — Application used within KStars for remote control of telescopes and related hardware via INDI.

**Stellarium** — Sky charts, has INDI plugin.

**Other INDI** — Many more applications work with INDI.

**INDIGO** — Positions itself as a next-generation INDI (?).

Using INDI with KStars and Ekos on a Sky-Watcher or Celestron telescope mount is a known working solution.

### 5.5.2 Antenna Tracking Software

At present, for the SatNOGS network RF ground stations, hamlib is typically used for tracking, if directional antennas are used. Hamlib was originally created for amateur radio equipment, but has expanded to control many more devices.

**hamlib** — Antenna rotator control.

I don't know of anyone prototyping a satellite tracking mount with hamlib, but it should be possible using the "leap frog" method, perhaps on the same mount as existing SatNOGS antennas. The mount may not be steady enough.

### 5.5.3 Camera Tracking Software

There are applications for using PTZ control of cameras, such as used in "security" cameras.

**motion** — PTZ camera control. XXX check

I don't know of any of these being used in satellite tracking prototypes at present.

## 5.6 GNSS with gpsd

GNSS is used for time synchronization and for (somewhat) accurately determining the location of the observation.

GNSS collectively includes the USA GPS, Europe's Galileo, Russia's GLONASS, and China's BDS, as well as other regional systems.

A basic, widely available COTS USB GNSS device with a basic (or no!) antenna plugged into the embedded system can get time and location accurate enough for the purposes here. See various U-Blox devices, for example.

In Debian GNSS service with a USB device can be provided by the gpsd application.

```
sudo apt install gpsd gpsd-tools
```

Edit the configuration file, /etc/default/gpsd like:

```
DEVICES="/dev/ttyACM0"
GPSD_OPTIONS="-Gn"
USBAUTO="false"
```

This will start a gpsd daemon listening on all interfaces, so it can be used by other computers on the network (such as if acquire and process are on different machines). Using the device /dev/ttyACM0 can be problematic if there are multiple /dev/ttyACM* devices. Perhaps something like this could be used in that case:

```
DEVICES="/dev/serial/by-id/usb-u-blox_AG_-_www.u-blox.com_u-blox_GNSS_receiver-if00"
```

Even with gpsd configuration listening on all IPs, don't think systemd won't do what it likes. So you may have to do:

```
systemctl edit --full gpsd.socket
```

And create a gpsd configuration like this (I have IPv6 disabled, in this case):

```
[Unit]
Description=GPS (Global Positioning System) Daemon Sockets

[Socket]
ListenStream=/run/gpsd.sock
# ListenStream=[::]:2947
ListenStream=0.0.0.0:2947
SocketMode=0600
BindIPv6Only=no

[Install]
WantedBy=sockets.target
```

(Re)start:

```
sudo systemctl restart gpsd.service gpsd.socket
```

Test all is good:

```
gpsmon -n
# or
cgps -u m
```

Enable to start on boot:

```
sudo systemctl enable gpsd.service gpsd.socket
```

See below for sample output from `cgps`.

```
                                Seen 39/Used 28
Time:        2022-09-02T00:40:28.000Z (18) GNSS   PRN  Elev   Azim   SNR Use
Latitude:          40.56896250 N            GP  1    1  24.0  237.0  44.0  Y
Longitude:        105.22600420 W            GP  3    3  63.0  297.0  50.0  Y
Alt (HAE, MSL):   1848.352,   1869.637 m    GP  4    4  33.0  292.0  45.0  Y
Speed:               0.05 km/h              GP 16   16  21.0  162.0  36.0  Y
Track (true, var):    85.1,   8.2     deg   GP 21   21  13.0  218.0  40.0  Y
Climb:               0.00 m/min             GP 22   22  40.0   77.0  18.0  Y
Status:          3D DGPS FIX (29 secs)      GP 26   26  43.0  140.0  46.0  Y
Long Err  (XDOP, EPX):  0.40, +/-  1.5 m    GP 31   31  57.0   49.0  39.0  Y
Lat Err   (YDOP, EPY):  0.36, +/-  1.3 m    SB131   44  42.0  198.0  44.0  Y
Alt Err   (VDOP, EPV):  0.90, +/-  0.7 m    SB133   46  37.0  214.0  41.0  Y
2D Err    (HDOP, CEP):  0.51, +/-  0.5 m    GL  6   70  48.0   99.0  21.0  Y
3D Err    (PDOP, SEP):  1.04, +/-  5.1 m    GL  7   71  62.0  352.0  51.0  Y
Time Err  (TDOP):        0.57               GL  8   72  19.0  314.0  47.0  Y
Geo Err   (GDOP):        1.18               GL  9   73  71.0  149.0  45.0  Y
ECEF X, VX:    -1274598.340 m   -0.010 m/s  GL 10   74  28.0  194.0  29.0  Y
ECEF Y, VY:    -4682900.030 m   -0.010 m/s  GL 23   87  13.0  305.0  48.0  Y
ECEF Z, VZ:     4127382.070 m    0.020 m/s  GA  2  302  31.0  168.0  40.0  Y
Speed Err (EPS):       +/-  0.4 km/h       GA  3  303  72.0   40.0  39.0  Y
Track Err (EPD):        n/a                GA  8  308  40.0  300.0  47.0  Y
Time offset:           -0.005160195 s      GA 13  313  22.0  285.0  43.0  Y
Grid Square:            DN70jn26           GA 15  315  17.0  232.0  43.0  Y
                       More...
```

## 5.7 NTP

With all the above, time still needs to be configured. Configuring gpsd isn't enough for correct system time. The main system on the Internet used for time synchronization is NTP. In Debian there are a few options for NTP. The best is to use a hardware GNSS (GPS), with PPS for improved accuracy. The easiest is to just use NTP.

All systems in the pipeline need to have the correct time and location. It is best if they all pull from the same NTP server, or even better than best if they all run GNSS hardware with PPS enabled.

Some quick and dirty time synchronization commands. The `ntpd` daemon can have slow startup synchronization sometimes. If so, perhaps run like this:

```
# Get the daemon out of the way.
sudo systemctl stop ntp

# Get time from someone, MIT in this case:
sudo ntpdate time.mit.edu

# OR

# Get time from local server, such as router (made up IP):
sudo ntpdate 192.168.1.1

# If above complains, to do an initial sync at boot (force ntpdate):
sudo ntpdate -b 192.168.1.1

# Restart daemon
sudo systemctl start ntp
```

Go through each of the systems used in the toolchain and make sure they all have accurate time or processing will be off.

## 5.8 KStars with Tracking Mount

KStars is an application that has been used for years in the astronomy community. It is well established, well supported, and included in Debian.

KStars has multiple uses for imaging satellites. One main use is to control a telescope tracking mount, such as the hardware described in section 4.7, page 40.

For the purposes here, described below will be using KStars with a Sky-Watcher tracking mount with INDI and Ekos. See figure 2.3, page 25 for a photo of the setup used with KStars below.

The telescope mount will be used in sidereal tracking mode, where the stars will appear as "points", and the satellites will appear as trails. The mount is not used in a satellite tracking mode. Sidereal is the "standard" tracking mode of telescopes.

KStars is the "main" application, but it depends on other key parts. INDI is the protocol that KStars uses for telescope control. INDI itself is a collection of applications. While KStars has the main sky chart and Ekos is launched within it, the actual mount control is done with the Ekos application. While it may sound complex, all of this is set up pretty easily in Debian.

```
# Quick and dirty from memory, something like:
sudo apt update
sudo apt install kstars indi-bin indi-eqmod indi-gpsd
```

KStars has a sky chart, as can be see in figure 5.3, page 57. When mount control is functioning, a location on the sky chart, such as a star, can be clicked on and the mount will GoTo that location and optionally track it. Using this, a telescope mount can be used to easily point the camera at a location and track it to observe satellites. It should also provide a superior FITS file for extracting data than using a static mount with star trails (XXX made up).

To use a telescope tracking mount for use with `stvid`, the following steps need to be performed in roughly this order, assuming everything has been configured. An overview of steps:

1. Physically mount all hardware.
2. Plug in and power everything up.
3. Confirm all hardware looks ok (e.g. `lsusb`).
4. Confirm GNSS time and location are ok (e.g. `cgps -u m`). See section 5.6, page 52.
5. Confirm time is correct on all systems in the toolchain (e.g. `date`). See section 5.7, page 53, for information about NTP.
6. Run camera configuration script (e.g. `v4l2-ctl` commands).
7. Start `indiserver` on the embedded system, using scripts to include a camera (such as `indi_v4l2_ccd`.
8. Start KStars on the workstation.
9. Launch Ekos within KStars, under `Tools`.
10. In Ekos, select a configuration with the EQ Mount for the Sky-Watcher, and `V4L2` for the CCD, which will work with The Imaging Source camera used in this example. Alternatively, the ZWO ASI could be used with a similar configuration.
11. The Ekos configuration should also be set to use the remote `indiserver` IP address of the embedded system USB connected to the Sky-Watcher mount.
12. Hit the start button to start Ekos/INDI.
13. On the screen that pops up, confirm all the tabs are good.
14. Check the last configuration tab for the camera, it often resets the size.
15. Hit close on the screen with the INDI devices.
16. Click the mount icon, and start tracking.
17. Perhaps do some focusing... XXX
18. Click the solver icon.
19. Click the Sync radio button.
20. Click Capture and Solve. XXX what is it.
21. If everything is tracking happily, good.
22. If not, do all the alignment steps.
23. When alignment is good and tracking is accurate, stop Ekos and close it.

24. Stop the `indiserver` running on the embedded system.
25. Start the `indiserver` on the embedded system, but without using a camera (e.g. remove `indi_v4l2_ccd`.
26. Select the INDI configuration with a remote `indiserver`, the EQ Mount, and the Simulated CCD.
27. Hit start in Ekos to get INDI connections going.
28. Confirm all is ok in hardware tabs, then hit close.
29. Now in the KStars sky chart window there is control of the mount without interfering with the camera.
30. Start `stvid`. See XXX for more info.
31. When done capturing that part of the sky with `stvid`, stop `stvid`
32. Go to the KStars sky chart and right-click on the new location, and slew to it.
33. Start `stvid` again, pointing at the new location.
34. Repeat the last few steps each time a new sky location is desired.

If the camera and mount are connected to the embedded system OK, it will look like below, in this case with The Imaging Source camera and Sky-Watcher mount:

```
jebba@odroid-01:~$ lsusb
Bus 002 Device 003: ID 199e:9004 The Imaging Source Europe GmbH DMK 33UX174
Bus 002 Device 002: ID 05e3:0620 Genesys Logic, Inc. GL3523 Hub
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 003: ID 067b:23d3 Prolific Technology, Inc. USB-Serial Controller
Bus 001 Device 002: ID 05e3:0610 Genesys Logic, Inc. Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

Note the `The Imaging Source Europe GmbH DMK 33UX174` device is the camera and the `Prolific Technology, Inc. USB-Serial Controller` is the Sky-Watcher telescope mount. Other views of the devices:

```
jebba@odroid-01:~$ ls -l /dev/bus/usb/002/003 /dev/bus/usb/001/003 /dev/video* /dev/ttyUSB0
crw-rw---- 1 root plugdev 189,   2 Sep  1 20:08 /dev/bus/usb/001/003
crw-rw-rw- 1 root video   189, 130 Sep  1 20:08 /dev/bus/usb/002/003
crw-rw---- 1 root dialout 188,   0 Sep  1 20:08 /dev/ttyUSB0
crw-rw---- 1 root video    81,   0 Sep  1 20:08 /dev/video0
crw-rw---- 1 root video    81,   1 Sep  1 20:08 /dev/video1
crw-rw---- 1 root video    81,   2 Sep  1 20:08 /dev/video2
```

Side note: you need to be in the `plugdev, video, dialout` groups. Side note: on the Odroid N2, there is another `/dev/video` device, so three devices are appearing. The The Imaging Source camera appears as two `/dev/video` devices.

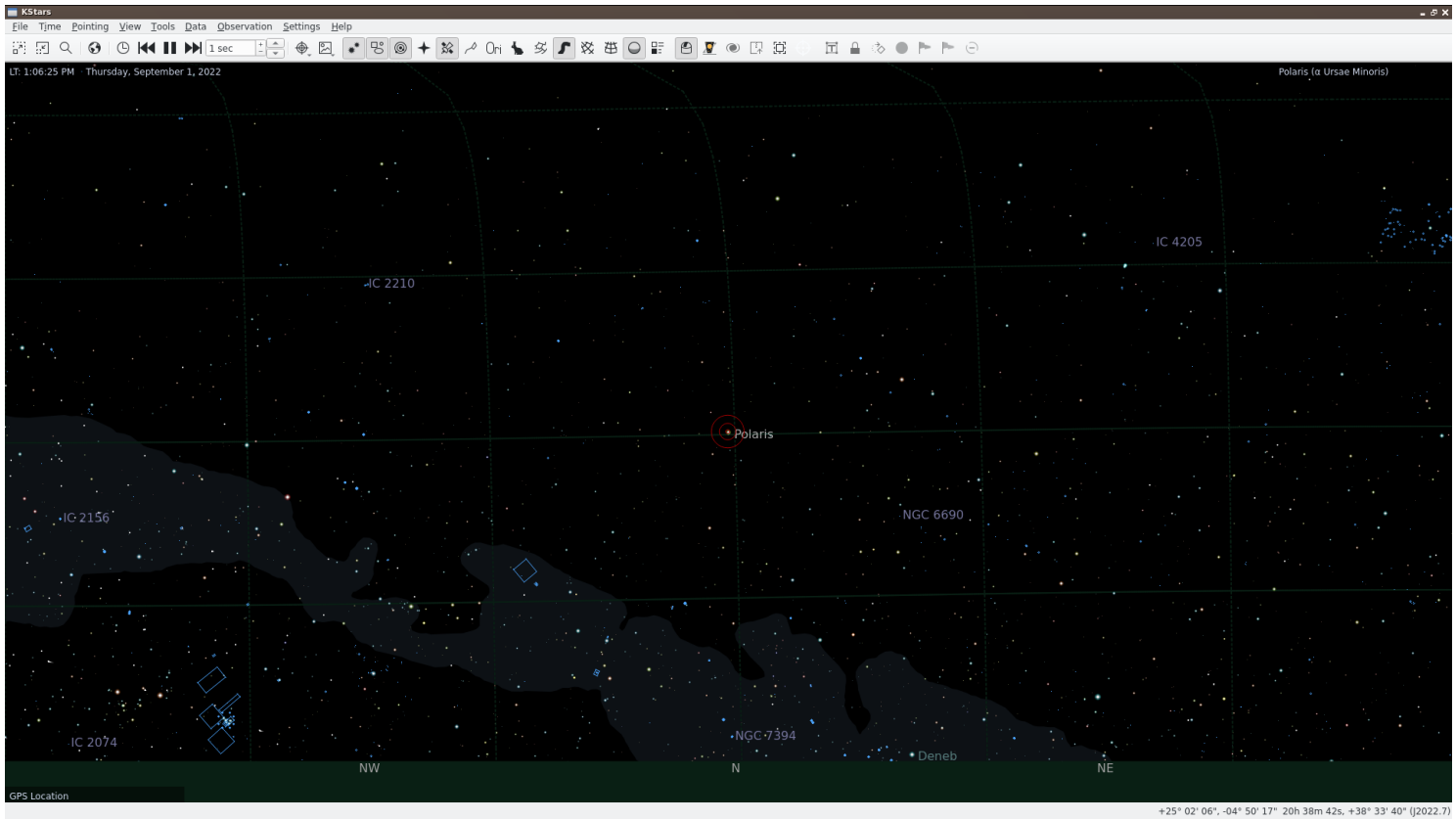



**Figure 5.3:** KStars sky chart, example screenshot.

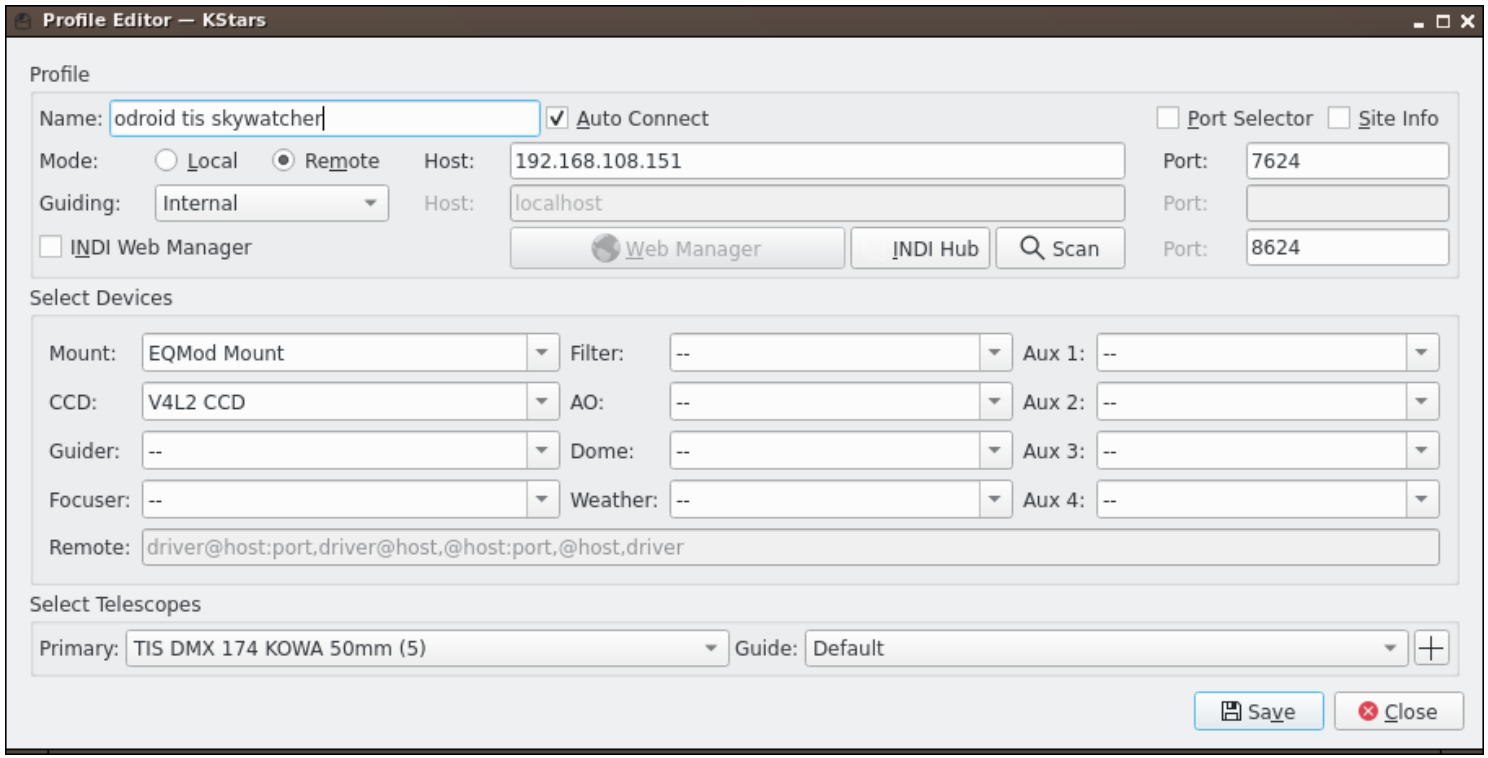



**Figure 5.4:** KStars using V4L2, example screenshot.

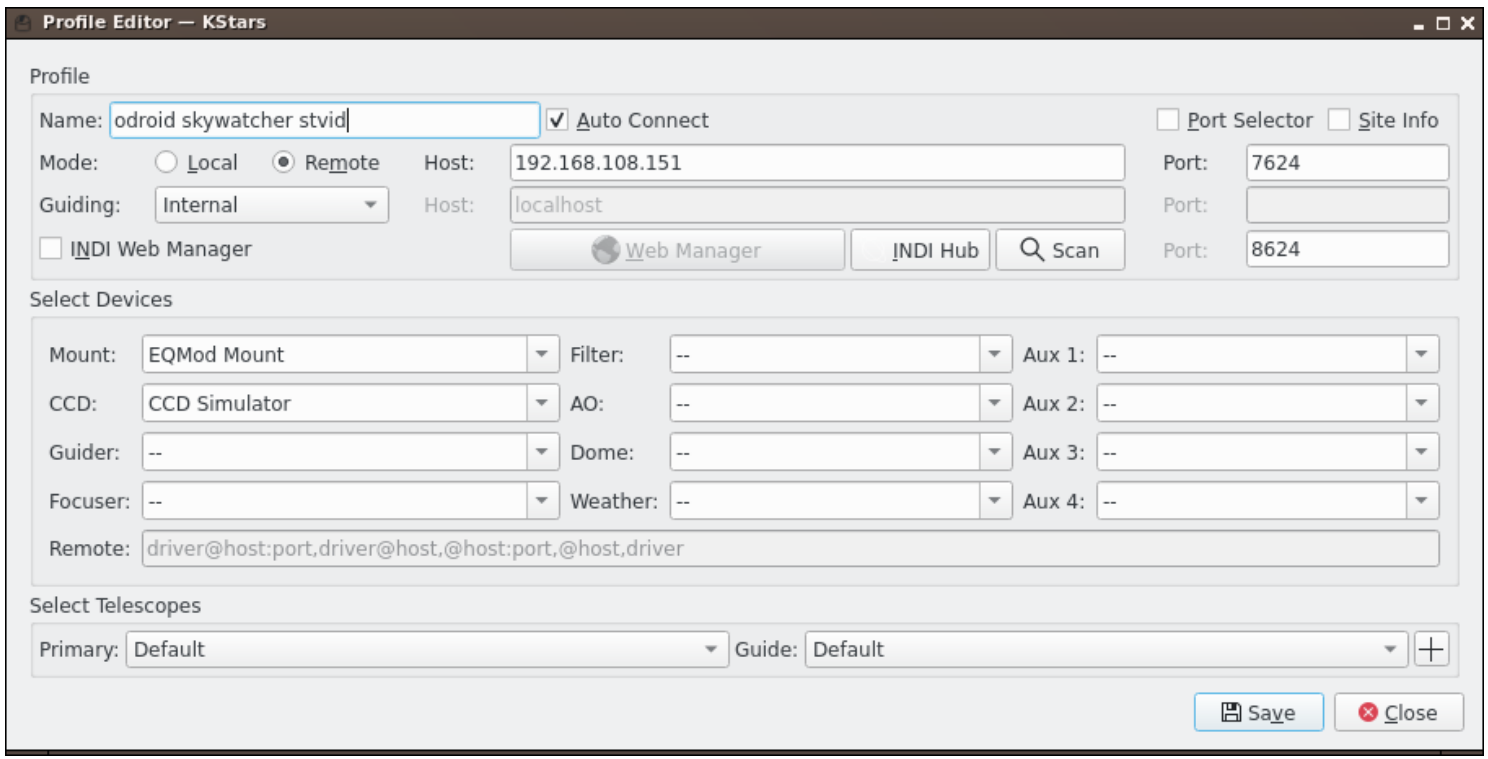



**Figure 5.5:** KStars using simulated CCD, example screenshot.

## 5.9 `stvid` with Docker

Daniel Ekman has created a Docker setup for easier installation and configuration.

The Docker source code repo is available here:

* `https://github.com/kng/stvid-docker`

The Docker itself is here:

* `https://hub.docker.com/repository/docker/knegge/stvid`

# Acquire Data

# Twinkle, Twinkle, Little Starlink

## 6.1 Overview of Operation

Below shows how to run a SatNOGS Optical ground station, after it has been set up and configured.

## 6.2 Setup with `stvid`

Acquire data with `acquire.py` from `stvid`. Enter `virtualenv`, if used:

```
cd stvid/
source env/bin/activate
```

Get the latest TLEs:

```
./update_tle.py
```

Acquire live data from camera.

```
cd stvid/
./acquire.py
```

## 6.3 `acquire.py` Output

Example output while running `acquire.py`.

```
2022-09-01 22:33:09,495 [MainThread  ] [INFO ]  Using config: configuration.ini
2022-09-01 22:33:09,495 [MainThread  ] [INFO ]  Test mode: True
2022-09-01 22:33:09,495 [MainThread  ] [INFO ]  Test duration: 3600s
2022-09-01 22:33:09,496 [MainThread  ] [INFO ]  Live mode: False
2022-09-01 22:33:09,517 [MainThread  ] [INFO ]  Starting data acquisition
2022-09-01 22:33:09,518 [MainThread  ] [INFO ]  Acquisition will end after
↪  2022-09-02T05:33:09.496
2022-09-01 22:33:09,620 [MainThread  ] [INFO ]  Storing files in
↪  /home/jebba/devel/spacecruft/stvid/data/obs/20220902_1/043309
2022-09-01 22:33:20,086 [MainThread  ] [DEBUG]  Captured z1
2022-09-01 22:33:20,086 [MainThread  ] [DEBUG]  Processing buffer 1
2022-09-01 22:33:21,842 [MainThread  ] [INFO ]  Compressed 2022-09-02T04:33:09.871.fits in
↪  1.75 sec
2022-09-01 22:33:21,842 [MainThread  ] [DEBUG]  Processed buffer 1
2022-09-01 22:33:30,171 [MainThread  ] [DEBUG]  Captured z2
2022-09-01 22:33:30,172 [MainThread  ] [DEBUG]  Processing buffer 2
2022-09-01 22:33:31,930 [MainThread  ] [INFO ]  Compressed 2022-09-02T04:33:20.136.fits in
↪  1.76 sec
```

**Figure 6.1:** Satellite images acquired by stvid.

# Plate Solvers

# Place from Pixels

## 7.1 Plate Solver

A "plate solver" will take an image of stars and detect the time and place of the picture. There are two main steps:

1. Extract stars from an image, such as a FITS file generated by `stvid`.
2. "Solve" the image of the stars in the image against vast databases in a star catalogue.

See below for a list of software tools that detect/extract stars from images (e.g. FITS).

- Source Extractor.
- KStars with Ekos adds `Internal SEP`.
- KStars with Ekos adds `Builtin method for solver`.

See below for a list of plate solvers that can be used.

- astrometry.net locally.
- astrometry.net Internet.
- astap.
- Watney.

## 7.2 Source Extractor

`Source Extractor`, formerly `SExtractor` and some other names, is an application to extract stars from images. The software has been widely used for many years.

`https://www.astromatic.net/software/sextractor`

## 7.3 astrometry.net

Astrometry.net is a website and software that can be run locally to solve a plate of stars that has been extracted from `Source Extractor`. XXX

NOTICE:
Using astrometry.net online requires an account with non-libre services. The astrometry.net software can be installed locally to avoid this.

## 7.4 Star Catalogues

To use a plate solver, you will need star catalogues. They can get large. The `stvid` application includes a basic star catalogue. XXX The `4200` index series is also recommended.

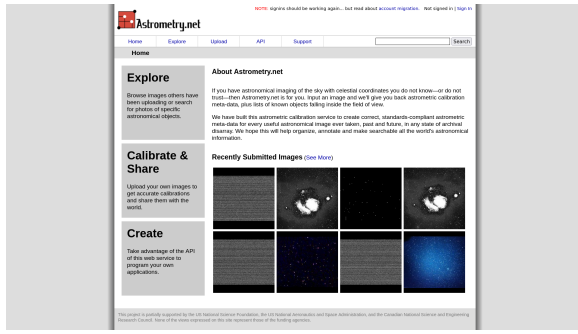



**Figure 7.1:** nova.astrometry.net website.

## 7.5 World Coordinate System

WCS info. XXX

## 7.6 SIP

SIP info. XXX

## 7.7 Plate Solving with `stvid`

Use the `process.py` scripts described in section 7.1, page 66, to solve plates with `stvid`.

# Detect

# Find Satellites

## 8.1 Satellite Detection with stvid

This is a description of satellite detection processes. To detect satellite in a FITS file using the stvid toolchain, run `process.py` or, if it exists, the `process_new.py` Python script.

Note the `stvid` application's `process.py` and `process_new.py` will perform both the detection and identification steps.

### 8.1.1 `process_new.py` Usage

This assumes you have installed `stvid` as shown in section 6.2, page 62. Adjust the dated directory used in this example to the actual directory name.

```
cd stvid/
# Maybe you need to update TLE
./update_tle.py
# Process directory
./process_new.py -c config_new.ini -d data/obs/20220825_1/020502/
```

### 8.1.2 `process.py` Usage

If there is no `process_new.py` it has likely been merged into `process.py`. Or perhaps, to use the "older" script it would be run as shown below. Note, the old and new configuration files are different and should be used with the correct corresponding Python script.

```
cd stvid/
./process.py -c configuration.ini -d data/obs/20220825_1/020502/
```

### 8.1.3 `process_new.py` Example Output

```
Computing astrometric calibration for
↪  /srv/obs/odroid-01/obs/20220902_1/043309/2022-09-02T04:33:09.871.fits
2022-09-02T04:33:09.871.fits 216.814874  31.756777  199/ 342  10.6  13.9  16.53 +-   0.69
48144 21 027BE  9999 G 20220902043314699 17 25 1431290+312311 37 S
51636 22 012Q   9999 G 20220902043314094 17 25 1413930+340692 37 S
File queue empty, waiting for new files...
2022-09-02T04:33:20.136.fits 216.815061  31.756834  198/ 345  10.8  13.2  16.53 +-   0.70
2022-09-02T04:33:30.223.fits 216.815420  31.756810  202/ 338  10.5  13.9  16.53 +-   0.70
90000 22 500A   9999 G 20220902043336020 17 25 1433368+375908 37 S
2022-09-02T04:33:40.315.fits 216.815570  31.756676  206/ 342  10.1  13.1  16.54 +-   0.70
```

## 8.2 AstroImageJ

Analyze data with applications, such as AstroImageJ.

FITS files are commonly used in astronomy.[1] AstroImageJ can read FITS format files. See 8.1, page 71, for a screenshot displaying a FITS file generated by `stvid`.

[1] Wikipedia contributors. *FITS — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=FITS&oldid=1091431488. [Online; accessed 28-August-2022]. 2022

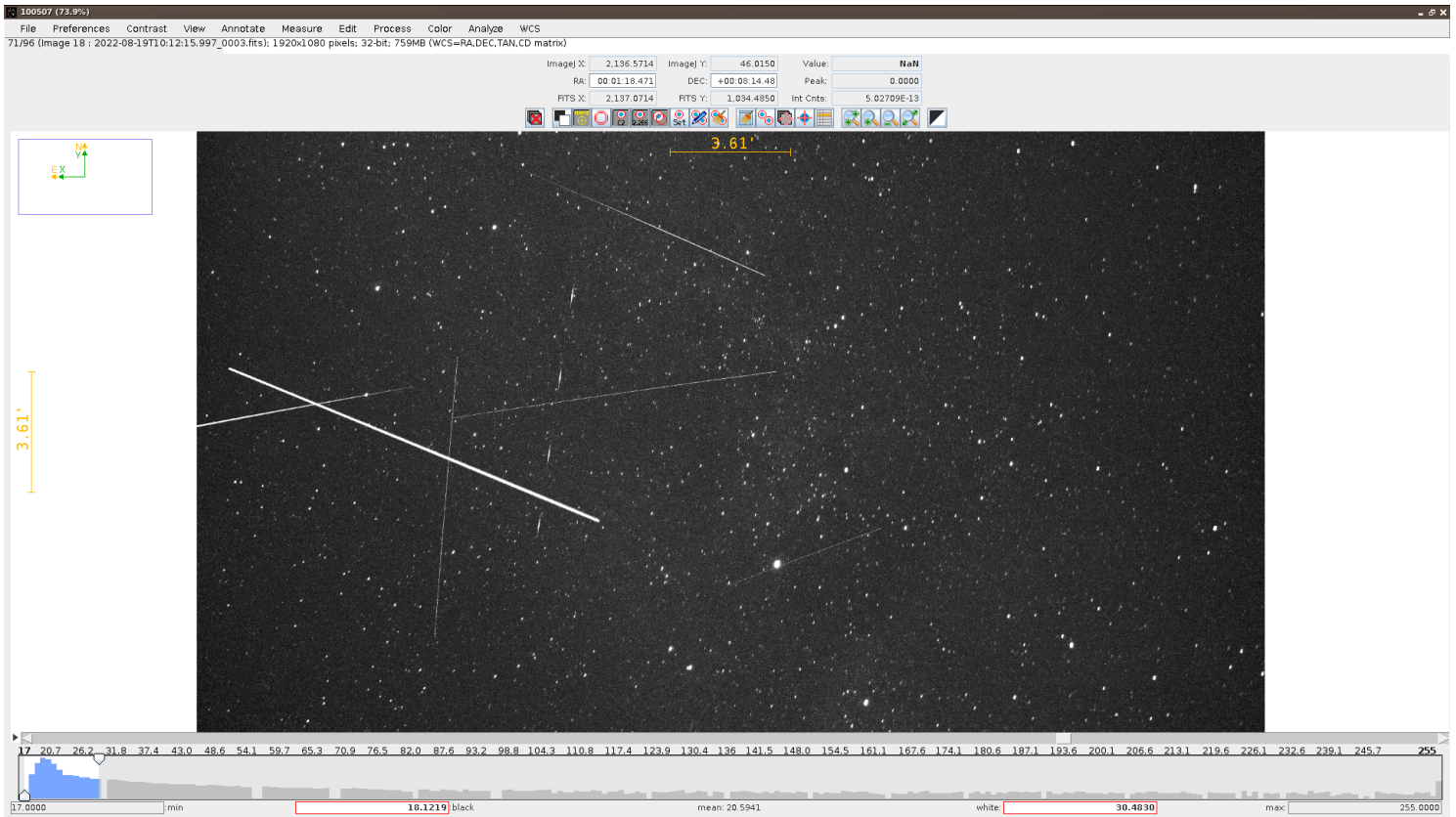



**Figure 8.1:** AstroImageJ screenshot viewing FITS file.

# Identify

# More Than Just Old Rocket Bodies

## 9.1 Overview of Satellite Identification

The LSF is developing a project called SIDLOC, for satellite identification and localization. With the addition of many more satellites in orbit there is a growing need for SSA.

## 9.2 `stvid`'s `process.py` for Identification

The best tool for identifying satellites from FITS files is the `process.py` script from `stvid`, available here:

* `https://github.com/cbassa/stvid`

My fork is here:

* `https://spacecruft.org/spacecruft/stvid`

## 9.3 Identifying Satellites with `satid`

The deprecated C application, `satid` from the `sattools` package can help identify satellites. See figure 9.1, page 75 for output from my Giza port of `satid`.[1]

* `https://github.com/cbassa/sattools`

My fork is here:

* `https://spacecruft.org/spacecruft/sattools`

## 9.4 Identification with stvid

See section 8.1, page 70 for details on identification with stvid. The detection and identification steps are both done in one script.

See figure 9.2, page 76 for an example of from stvid automated processing of an input FITS file. The output image is in PNG format. stvid will also output other files with results, such as CSV files.

## 9.5 Review of Images

See the following subsections for example output from stvid.

### 9.5.1 Unidentified Satellites

When stvid runs the `process.py` (or new) script and it encounters a satellite it cannot identify, it gives it the NORAD ID `90000`. If more unidentified satellites are detected in the same image, each detection is incremented by one.

See figure 9.3, page 77, for an example image, with three unidentified satellites. One is on the left, the other two on the right, next to each other.

See figures 9.4, 9.5, and 9.6, pages 78, 79, and 80, to see an example of `stvid` labelling three identified satellites with NORAD IDs `90000`, `90001`, `90002`.

[1] `https://spacecruft.org/spacecruft/sattools/media/branch/spacecruft/img/satid-giza-3.png`

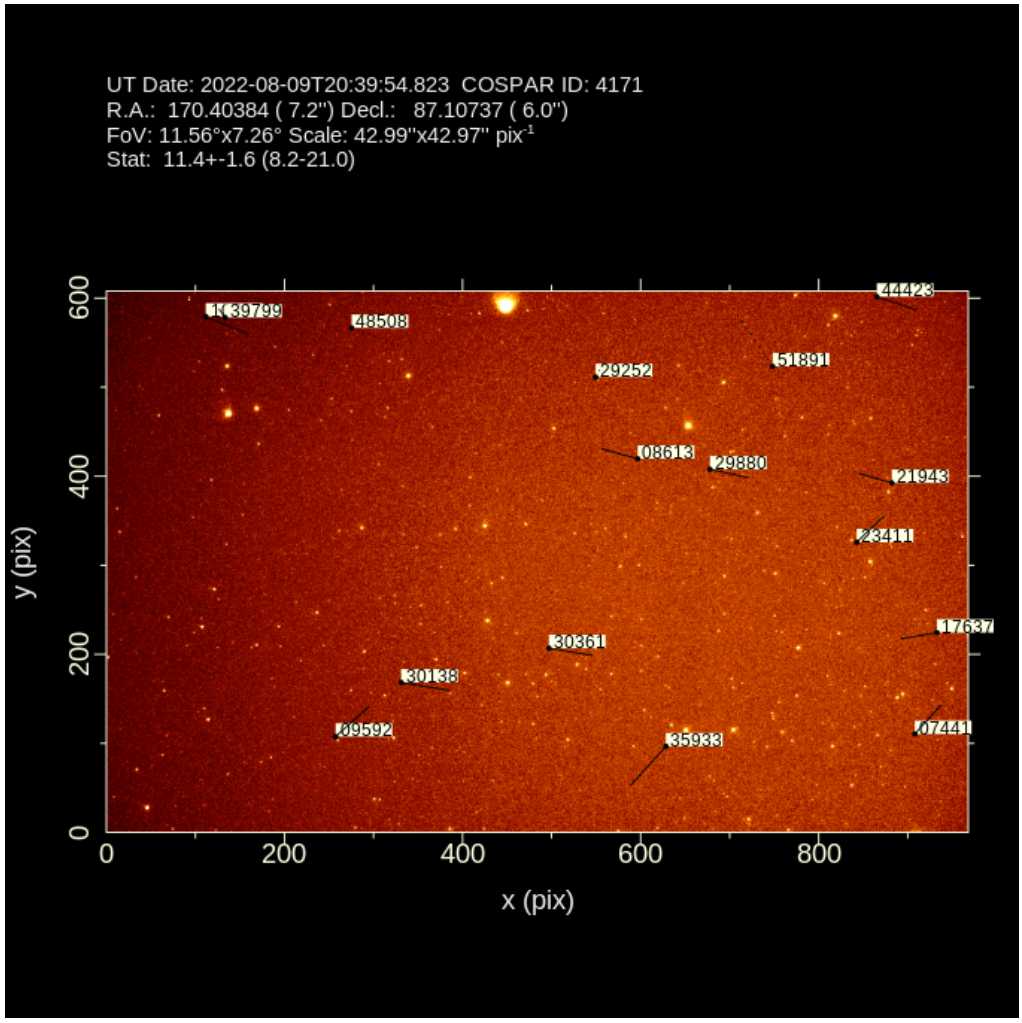



**Figure 9.1:** `satid` example.

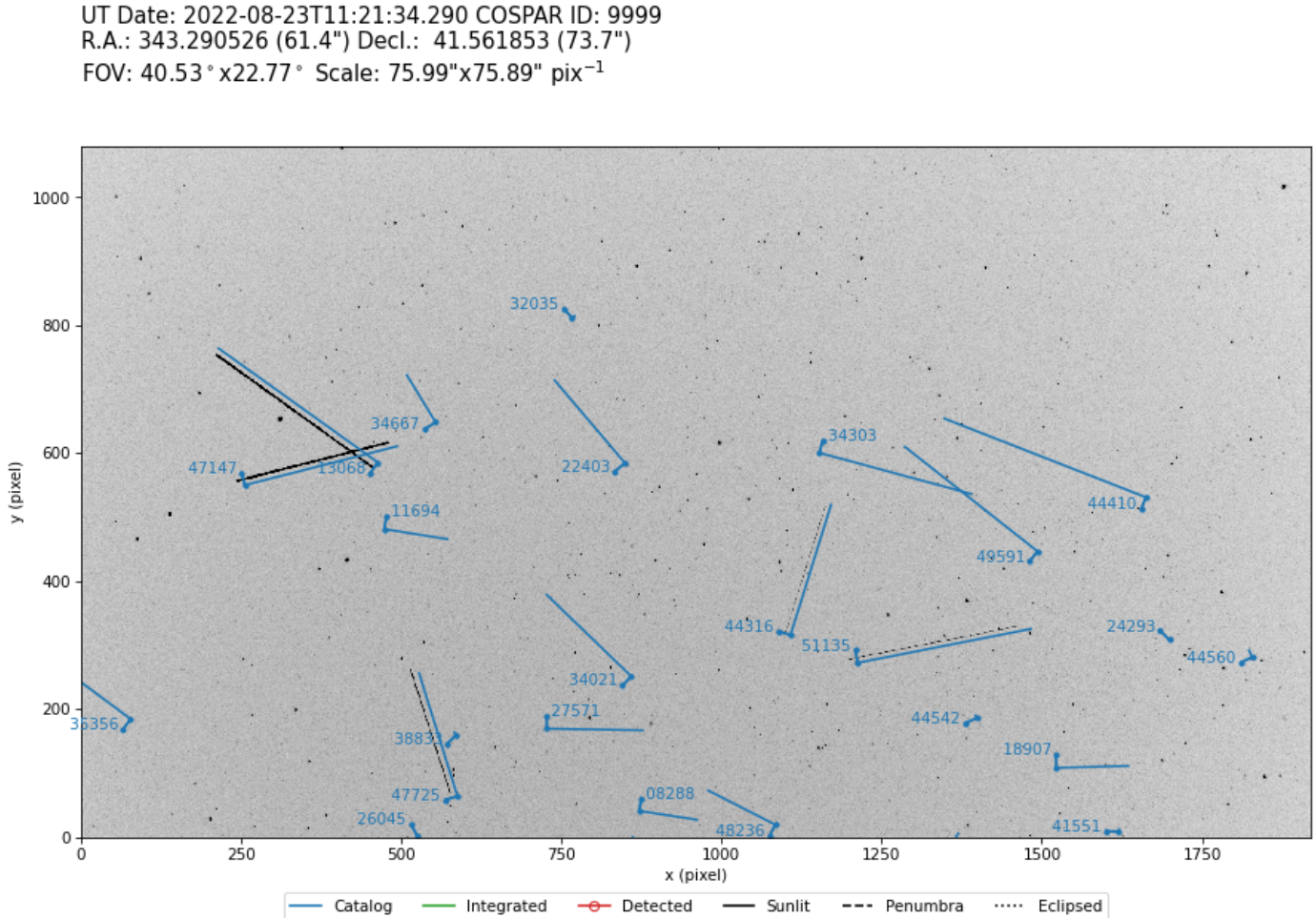



**Figure 9.2:** Example satellite image processed by `stvid`.

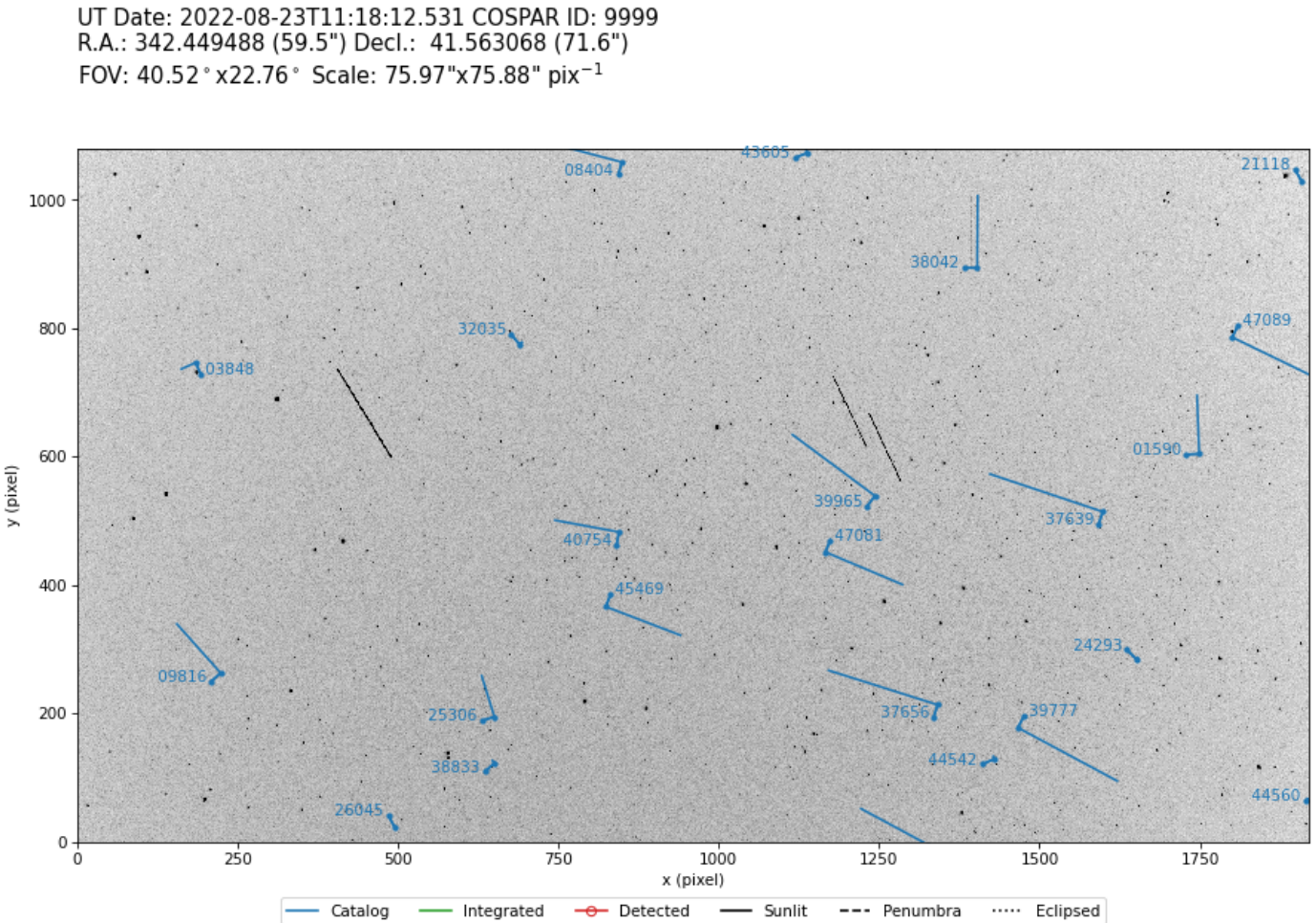



**Figure 9.3:** stvid processed image, with three unidentified satellites.

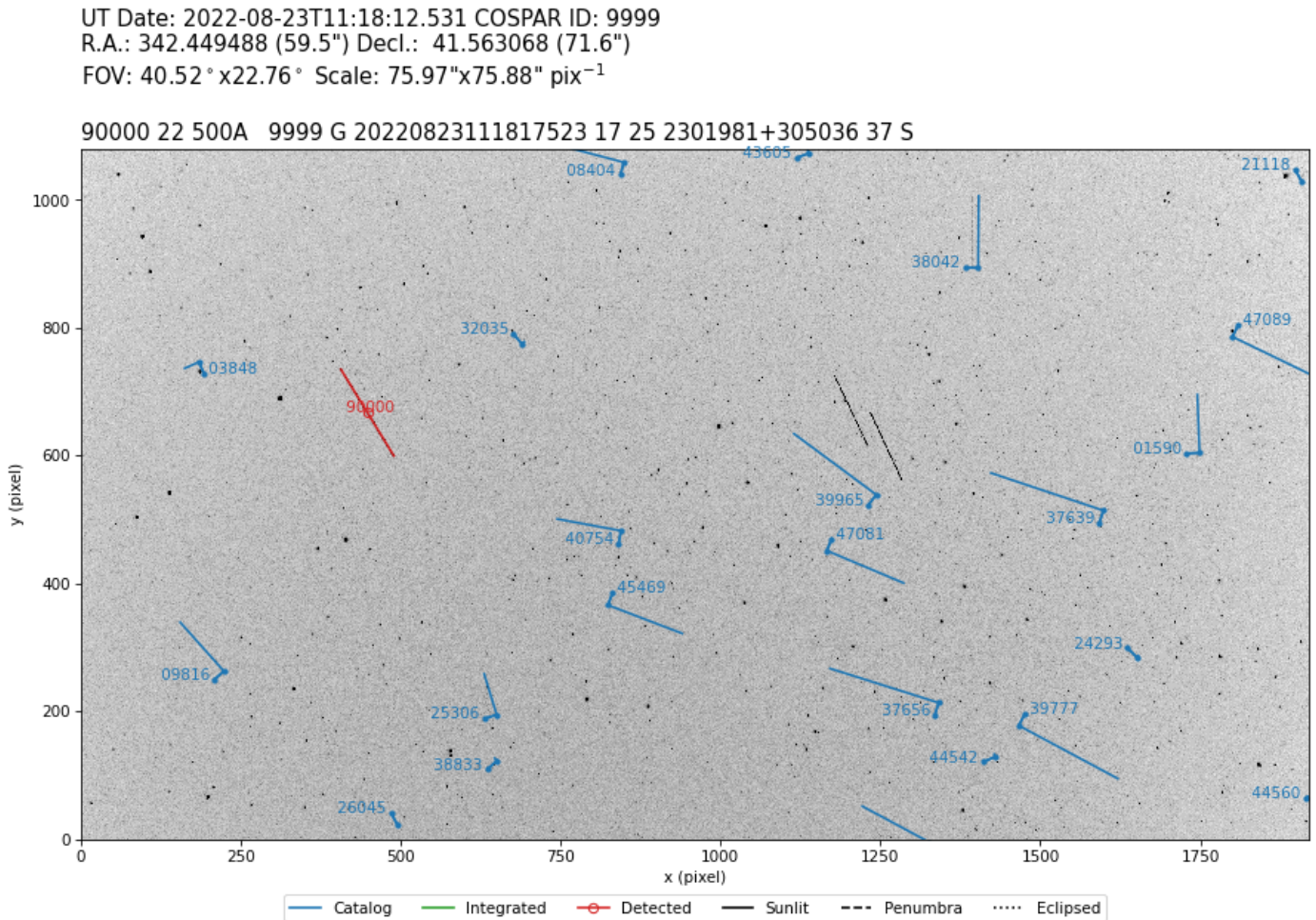



**Figure 9.4:** stvid processed image, with three unidentified satellites, the first labeled with NORAD ID 90000.

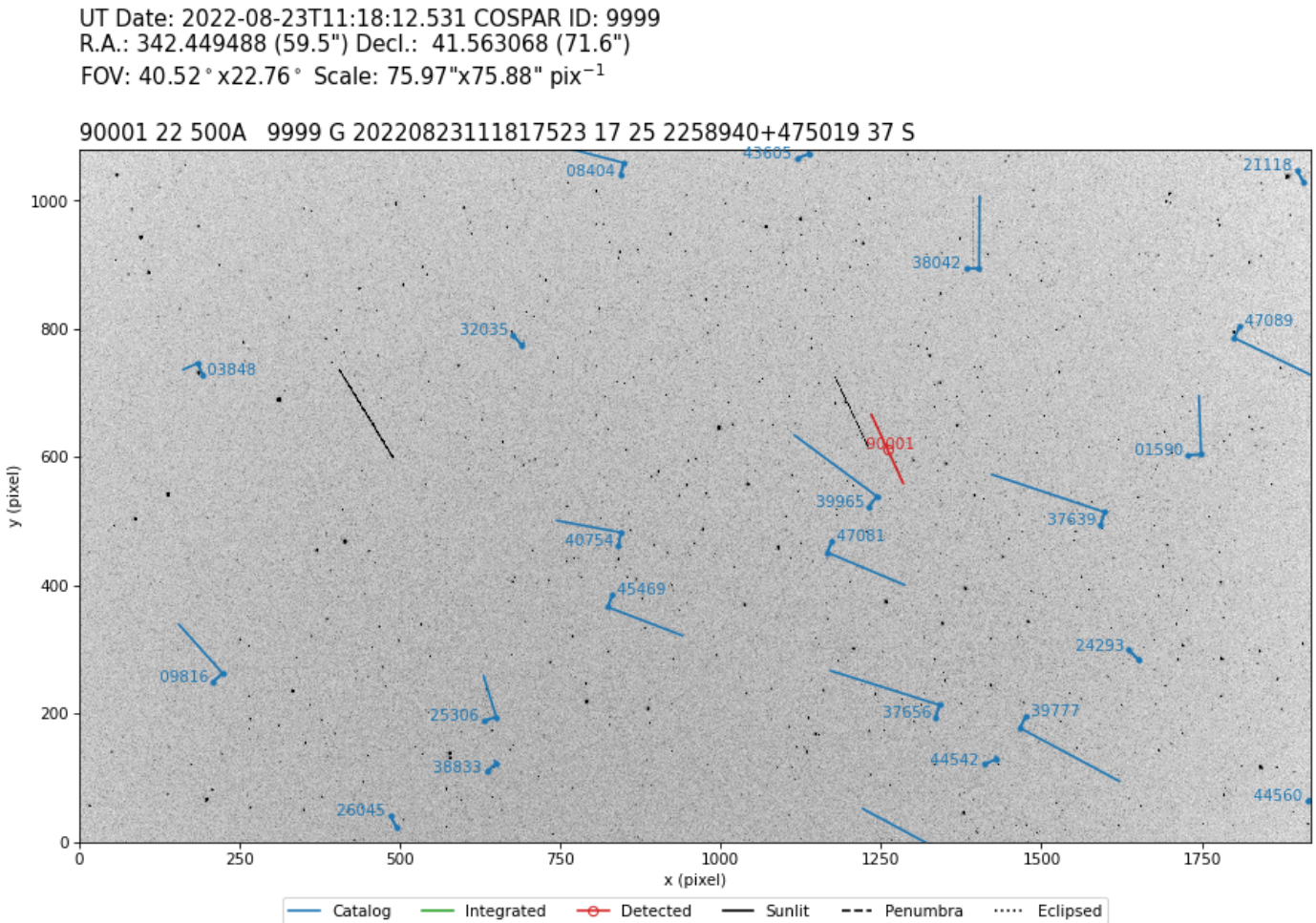



**Figure 9.5:** stvid processed image, with three unidentified satellites, the second labeled with NORAD ID 90001.

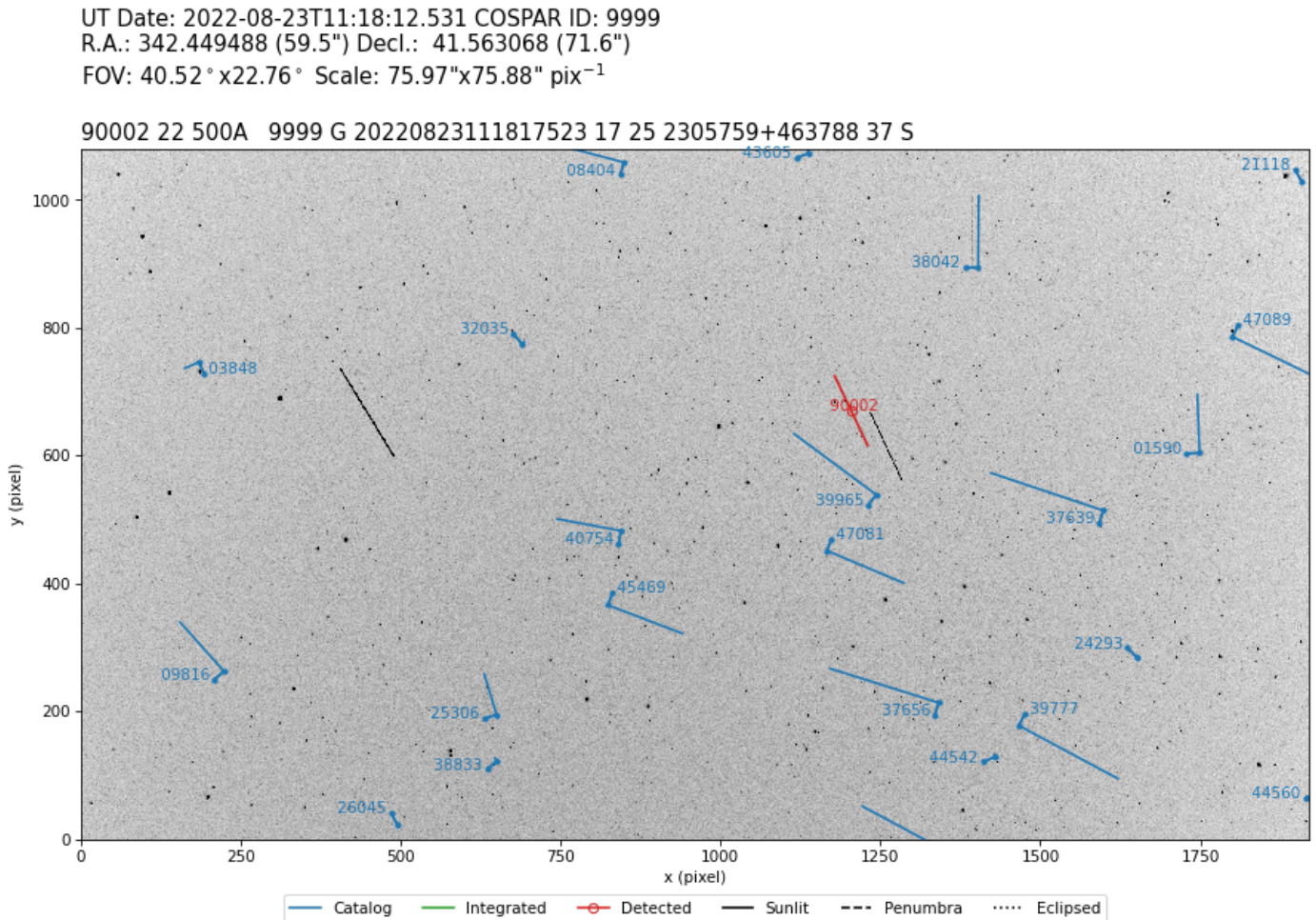



**Figure 9.6:** stvid processed image, with three unidentified satellites, the third labeled with NORAD ID 90002.

### 9.5.2 Airplanes

Airplanes are noise in the data.

Airplanes will appear not infrequently in images. See figure 9.7, page 82 for an example with an airplane in the left side, forming two parallel trails from the aircrafts' lights.

In figure 9.8, page 83, is shown an example image with an airplane with a very distinct blinking light.

In figure 9.9, page 84, airplane solid parallel trails can be seen with blinking lights.

### 9.5.3 Configuration and Setup Errors in Images

See figure 9.10, page 85, for an example of an image processed by a mis-configured system. As can be seen, the plotted lines are offset from the actual images of satellites. This, and similar issues, can be addressed by checking:

- Recent TLEs on both the processing workstation and the embedded system.
- Correct, NTP (or better) synchronized time on on the processing workstation and the embedded system.
- Correct latitude, longitude, and altitude are set in configuration files, typically based on GNSS readings.

### 9.5.4 Image Processed with Bad TLE

In figure 9.11, page 86, can be seen a FITS file that has been processed by stvid, but with TLEs that are around two days old. As can be seen, the satellites don't quite line up where they should.

### 9.5.5 Image Acquired During Slew

See figured 9.12, page 87, for an example image that is bad due to slewing during acquisition.

### 9.5.6 Light Outside FoV Artifact

See figured 9.13, page 88, for an example of an artifact, visible crossing the left and center, generated by light outside the FoV, in this case the moon![2] This is also an example image that detects classified satellites. This is how the main upstream `stvid` works.

[2] Pierros in SatNOGS Optical Matrix

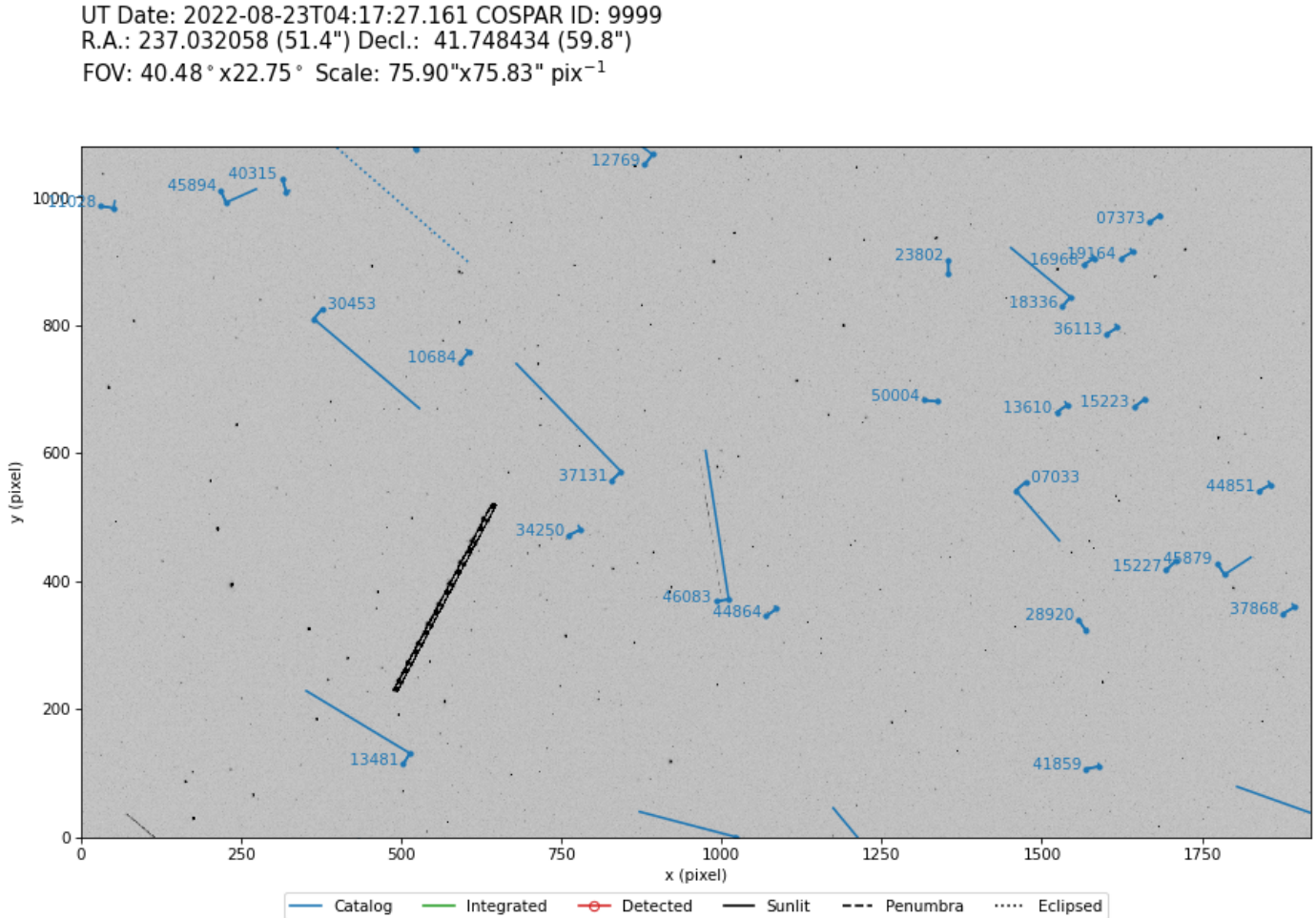



**Figure 9.7:** Airplane, bold left side, in stvid processed image.

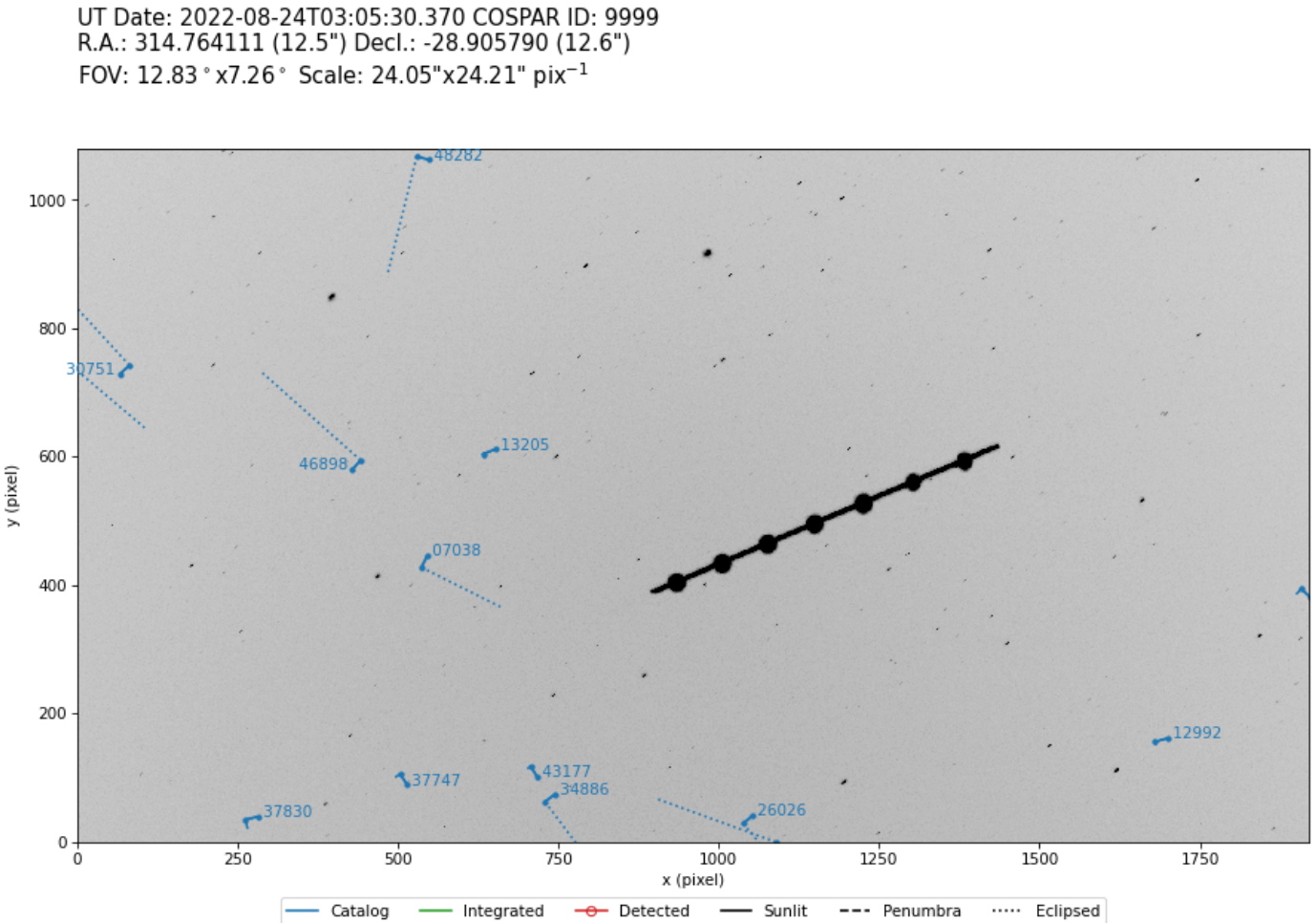



**Figure 9.8:** Airplane, blinking, in stvid processed image.

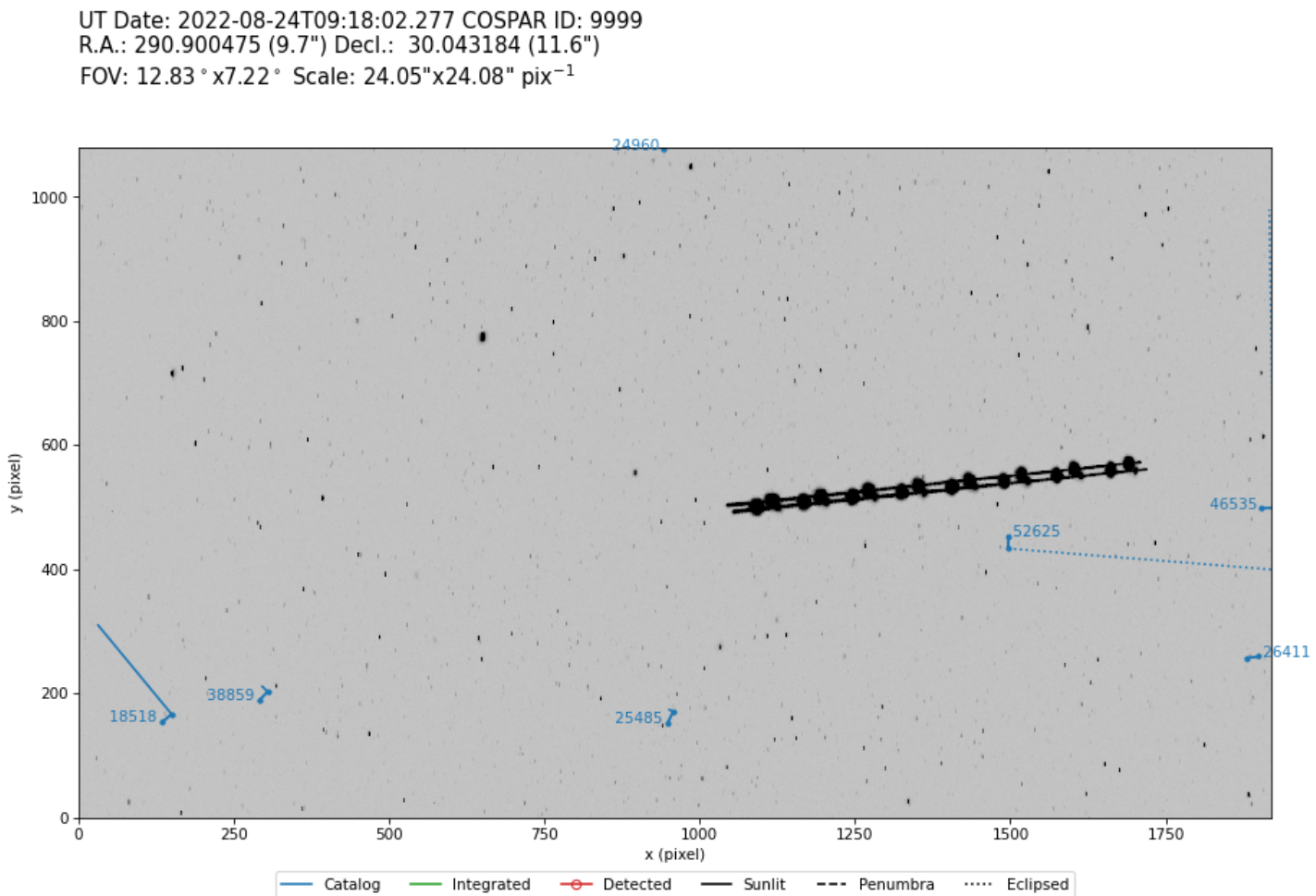



**Figure 9.9:** Airplane, two solid lights plus blinking, in stvid processed image.

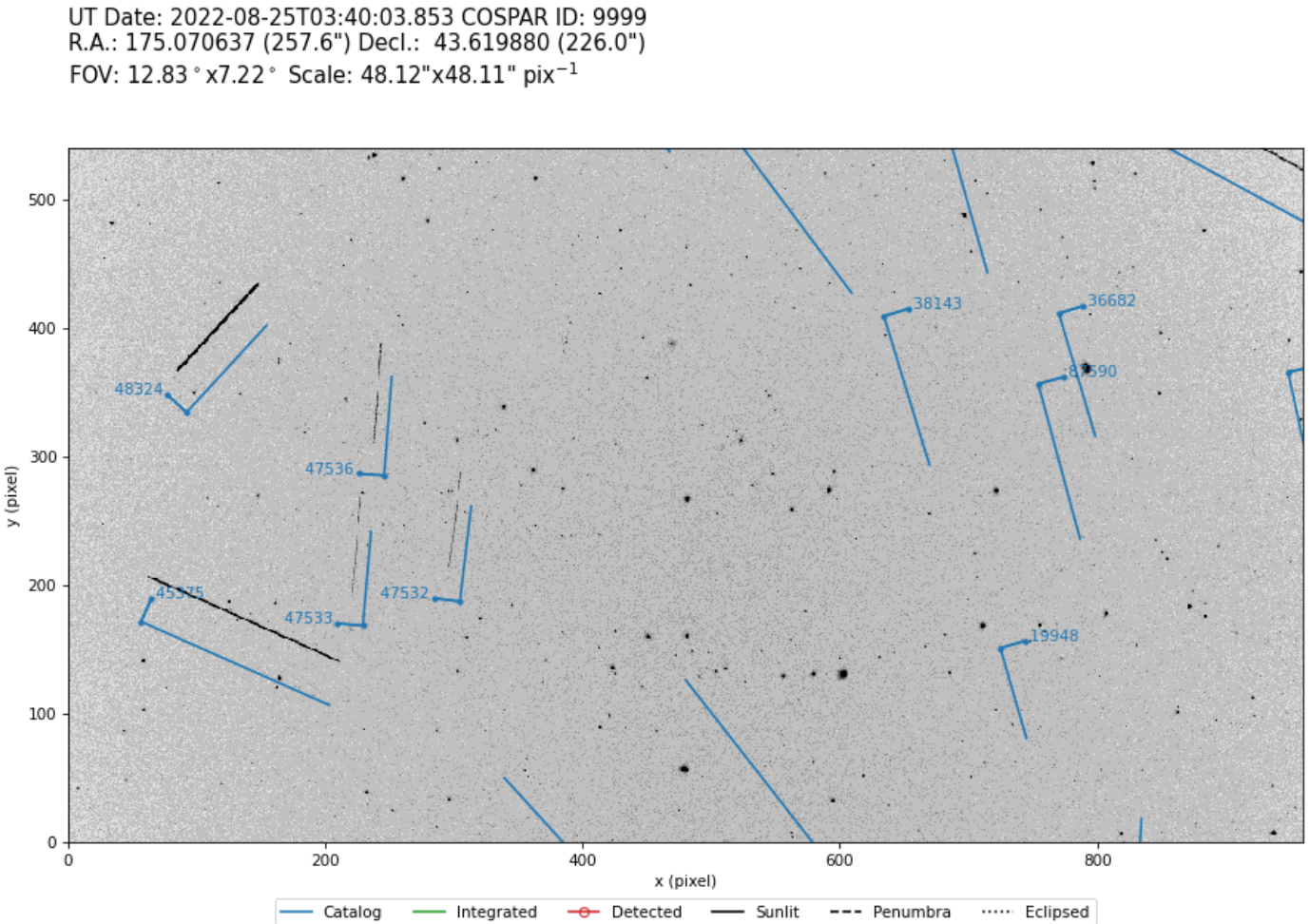



**Figure 9.10:** `stvid` processed image, showing mis-configured/setup system.

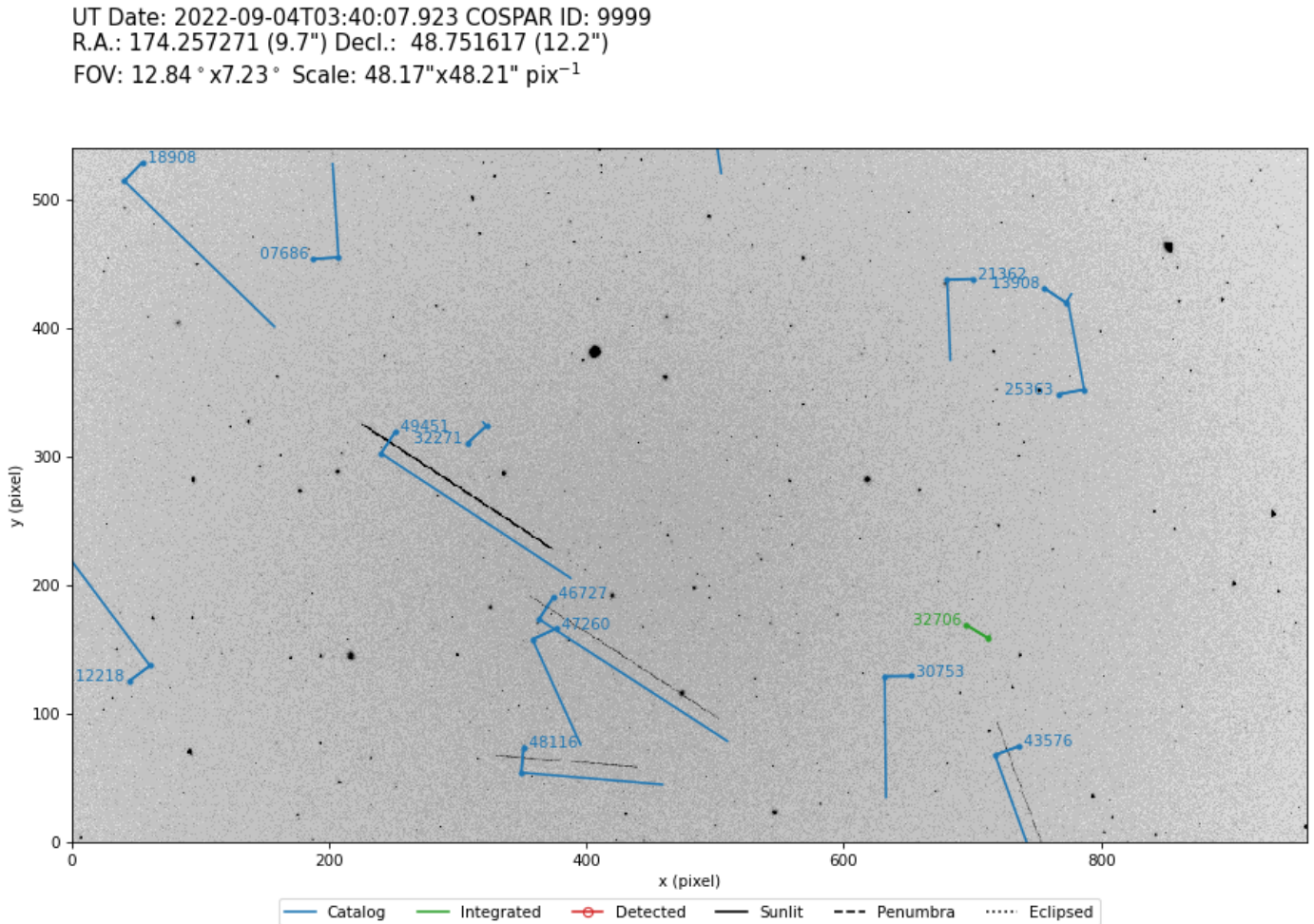



**Figure 9.11:** `stvid` processed image, alignment off due to use of old TLE.

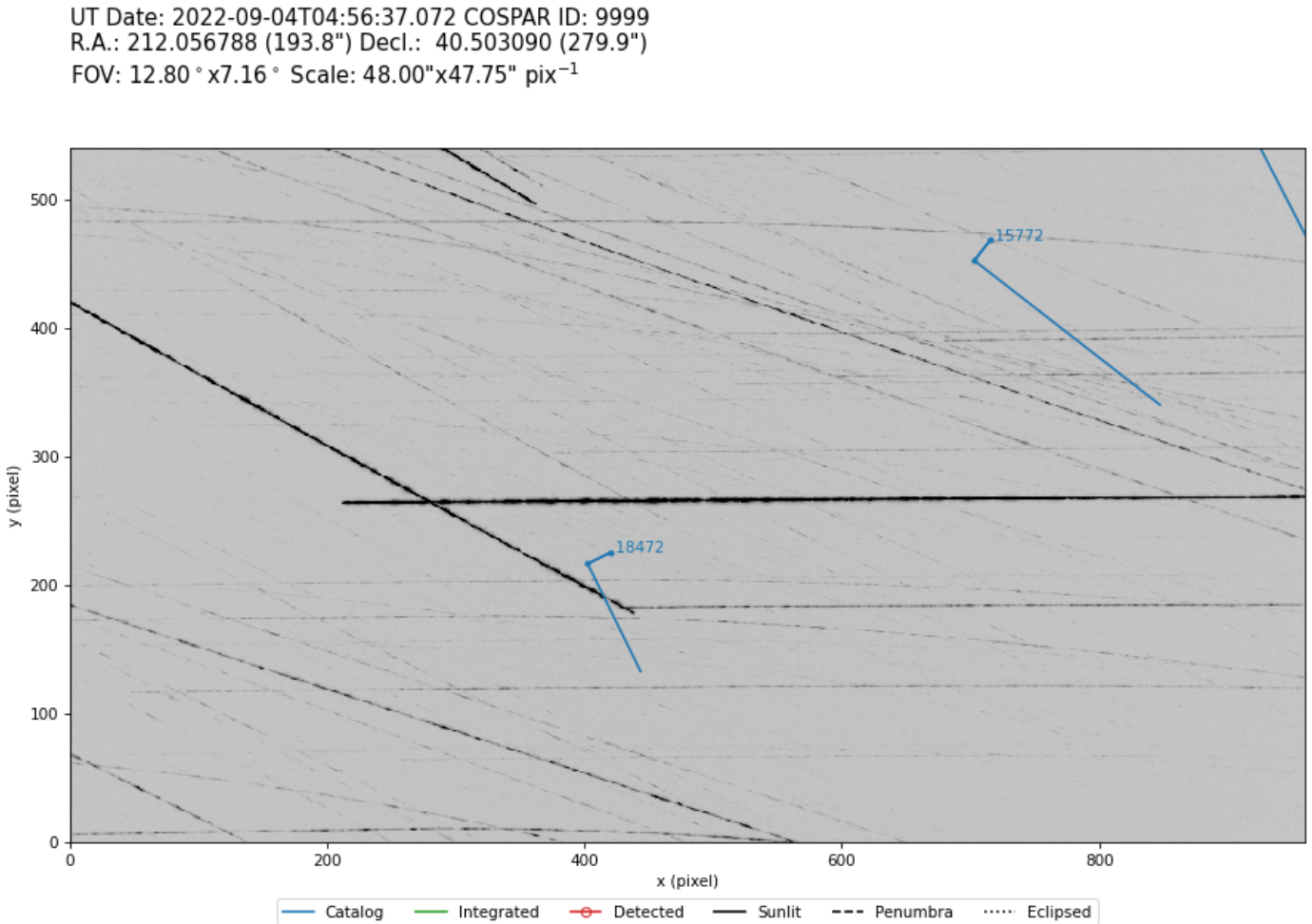



**Figure 9.12:** `stvid` processed image, trails due to slewing during acquire.

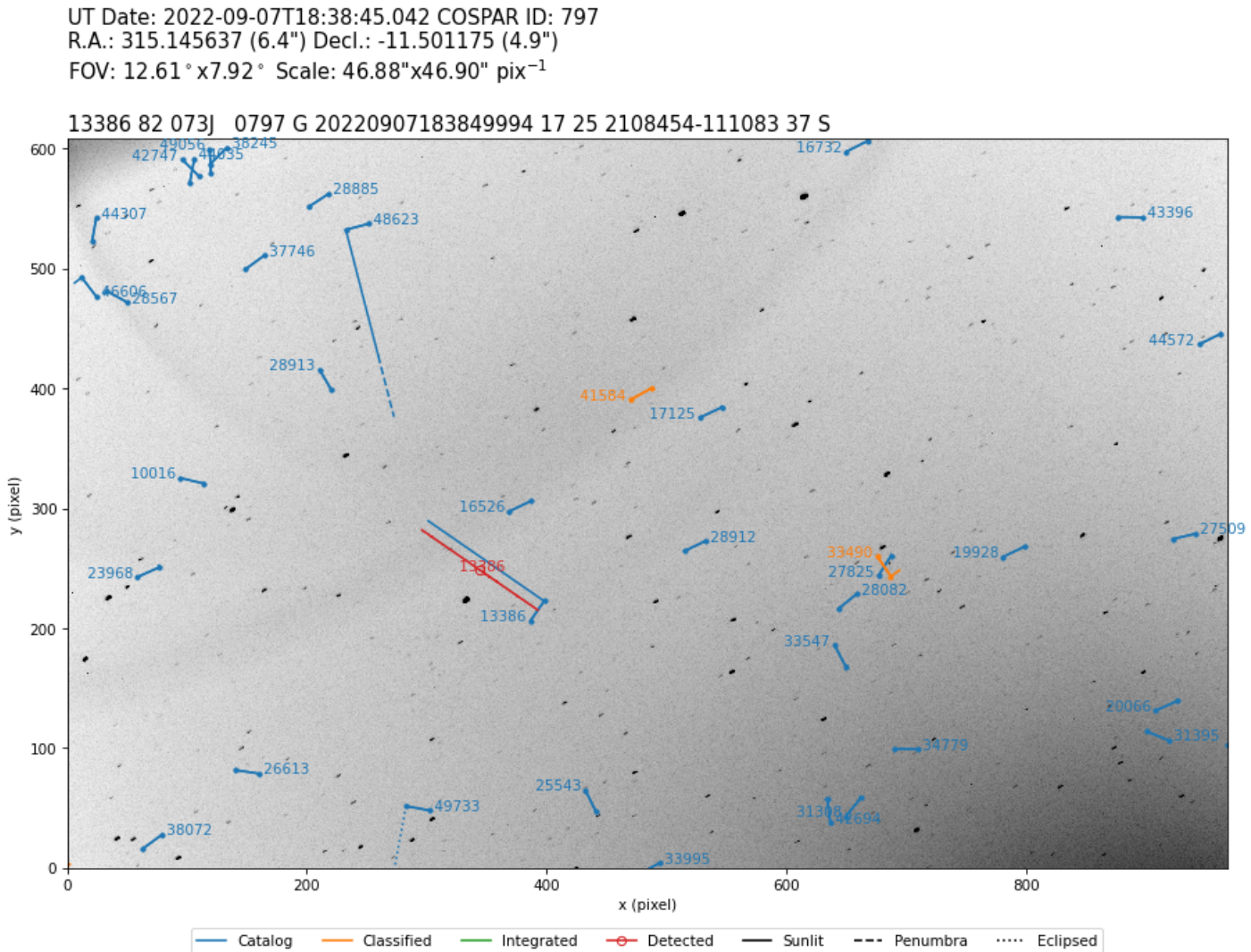



**Figure 9.13:** `stvid` processed image, with classified satellites and light artifact.

# Upload & Share

# Send Data to the SatNOGS Network

## 10.1 Push Data Upstream

The SatNOGS network isn't ready for optical data yet. So there is nothing that can be uploaded. This is under development.

## 10.2 File Extensions

The following file extensions are in use:

**.cal** — Cal

**.cat** — Cat

**.csv** — CSV

**.dat** — Data.

**.fits** — FITS.

**.png** — PNG.

## 10.3 Data Files

After the `stvid` acquisition and processing steps have been run, files such as below will be generated:

```
2022-08-23T04:16:26.633.fits
2022-08-23T04:16:26.633.fits.cal
2022-08-23T04:16:26.633.fits.cat
2022-08-23T04:16:26.633.fits.png
2022-08-23T04:16:26.633_48473_catalog.dat
2022-08-23T04:16:26.633_48473_catalog.png
2022-08-23T04:16:26.633_52718_catalog.dat
2022-08-23T04:16:26.633_52718_catalog.png
2022-08-23T04:16:26.633_90000_unid.dat
2022-08-23T04:16:26.633_90000_unid.png
2022-08-23T04:16:26.633_hough.csv
2022-08-23T04:16:26.633_predict.csv
2022-08-23T04:16:26.633_threshold.csv
```

### 10.3.1 `.fits` FITS Files

FITS files are the main files created during acquisition and are the main files used in processing. It has the data of the actual image taken with the camera. FITS files will be significantly larger than the other files, at around 8 to 35 megabytes each, depending on sample rate, frame size, and binning.

The `stvid` program creates it's customized FITS format file, which is a stack up of multiple images. I'll leave explanation of the `stvid` FITS file structure to the program's author, Cees Bassa:

With stvid, I typically store 10 seconds worth of video frames, 100 frames at 10 Hz frame rate or 250 frames at 25 Hz frame rate into a so-called "four frame". Here, for each pixel of the initial 100 or 250 frames, the mean, standard deviation, maximum and argmax are stored, each as its own frame. Objects that move will stand out because they will have a high (max - mean) / standard deviation, which allows me to detect trails. Once a trail is detected, the argmax frame tells me the frame number and hence the motion of the trail, and using the timestamps of the individual 100 or 250 frames, I can reconstruct the pixel position as a function of time during the 10 second exposure, using only 4 frames instead of the 100 or 250 individual video frames.

### 10.3.2 `.cal` CAL Files

Example `.cal` filename:
`2022-08-23T04:16:26.633.fits.cal`

Sample data from `.cal` file, just the top select lines of a 577 line file:

```
1172.876   11.776  7.1775 221.117561  45.185979  6.761
 786.524    1.817  6.0213 222.623526  37.272049  5.590
 838.954   21.772  8.0430 222.941344  38.403949  7.653
 872.352   25.919  8.3855 222.909215  39.103246  7.884
 692.542   37.123  8.2615 223.889916  35.455924  8.718
 738.829   47.126  8.2037 223.973142  36.430116  7.777
 691.964   52.656  7.7910 224.278547  35.490201  7.411
 951.962   53.159  8.0072 223.315748  40.816697  7.844
1209.704   60.030  6.0799 222.327791  46.116208  5.811
 893.057   66.086  7.1867 223.943996  39.648460  8.039
...
```

### 10.3.3 `.cat` CAT Files

`stvid` generates `.cat` files, generated from FITS files, such as:
`2022-08-23T04:16:26.633.fits.cat`

Sample data from `.cat` file, just the top select lines of a 1,033 line file:

```
#   1 X_IMAGE                Object position along x                                  [pixel]
#   2 Y_IMAGE                Object position along y                                  [pixel]
#   3 MAG_BEST               Best of MAG_AUTO and MAG_ISOCOR                          [mag]
#   4 FLUX_BEST              Best of FLUX_AUTO and FLUX_ISOCOR                        [count]
#   5 FLUX_MAX               Peak flux above background                               [count]
#   6 BACKGROUND             Background at centroid position                          [count]
  1172.8755     11.7756   7.1775     33.80787     11.09625     16.06616
    28.8250     11.4696   6.7443     50.38253      5.008112    15.81194
  1347.4933     10.4887   8.3649     11.32553      2.619246    16.04644
   269.9134      9.8019   8.4220     10.74506      1.56768     15.97937
   468.9823      7.9780   8.8072      7.535751     5.987358    16.02673
  1475.9681      6.6672   8.6115      9.024131     2.281431    16.03049
   847.0105      5.9795   8.1992     13.19223     14.31796     16.08833
   181.5139      5.5141   7.9473     16.63711      2.278276    15.95168
   460.1482      5.1577   8.5318      9.711239     2.795708    16.02688
  1559.9583      2.9245   7.5915     23.08835      6.981803     15.9909
```

```
   786.5237      1.8174   6.0213     98.05751     27.88283     16.09418
...
```

### 10.3.4 `.fits.png` FITS PNG Files

The `stvid` application can generate multiple PNG files, depending how many satellites it finds. In every case it creates at least one PNG file, with a name like:

`2022-08-23T04:16:26.633.fits.png`

In this image, as can be see in figure 10.1, page 92, no detected satellites are indicated. The blue lines are from the "Catalog", which in this case means TLEs. The blue lines are drawn whether no, a few, or many satellites are detected. They are generated by taking data from the plate solver about where the picture FoV is, and overlaying satellite orbits that are calculated from TLEs.

The main blue line is slightly offset from where the satellite is calculated to appear. The blue line forms a small L shape. The smaller segment indicates the area where the satellite may pass. Next to the small segment line is the NORAD ID for the satellite. If it is unknown, it will be given the number 90000 or larger.

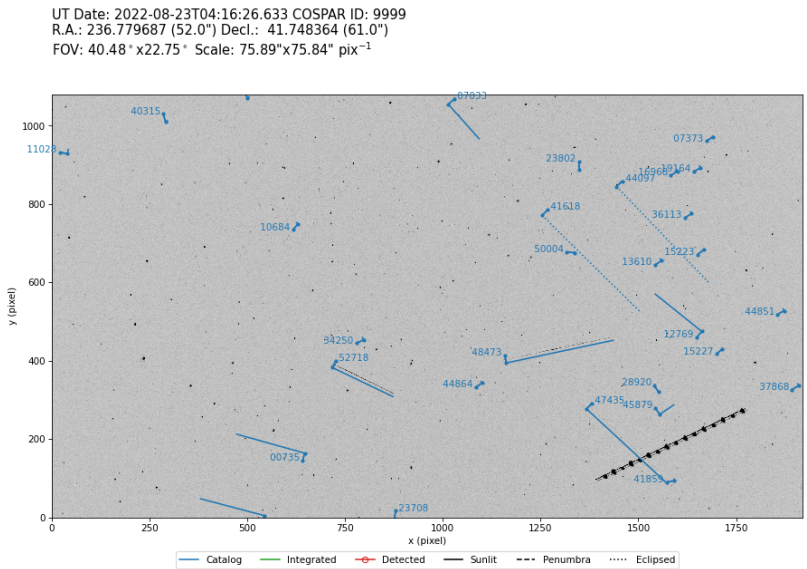



**Figure 10.1:** `stvid` PNG.

### 10.3.5 `catalog.png` catalog PNG Files

When `stvid` identifies a satellite, it creates a new PNG file for each satellite detected in the image. A file name example follows, with satellite NORAD ID 48473 identified, also shown in figure 10.2, page 93.

`2022-08-23T04:16:26.633_48473_catalog.png`

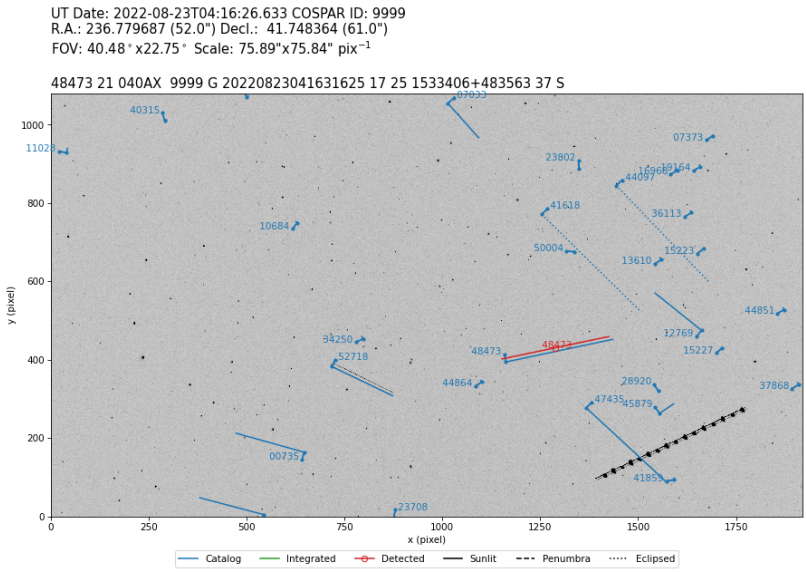



**Figure 10.2:** `stvid` PNG of the satellite trail of NORAD ID 48473 identified in red.

If multiple satellites are detected in the image, they will each get a PNG file. For the example image, a satellite with NORAD ID 52718 was also identified. The file name is:

`2022-08-23T04:16:26.633_52718_catalog.png`

### 10.3.6 `unid.png` Unidentified PNG Files

As shown previously, `stvid` will create a new PNG with a red line when it identifies a satellite. When `stvid` finds a trail it can't identify in the TLE catalogs, it gives it a NORAD ID starting with 90000, incrementing by one. See figure 10.4, page 95, for an example of `stvid` marking an unidentified trail red. In this case, the trail was created by an airplane, but `stvid` would do the same with an unidentified satellite trail.

### 10.3.7 `.dat` Data Files

For each trail detected as shown in previous sections, such as for which a PNG file was created, `stvid` will output a `.dat` file. The `.dat` filenames corresponding to the figures at 10.2, 10.3, and 10.4 are:

```
2022-08-23T04:16:26.633_48473_catalog.dat
2022-08-23T04:16:26.633_52718_catalog.dat
2022-08-23T04:16:26.633_90000_unid.dat
```

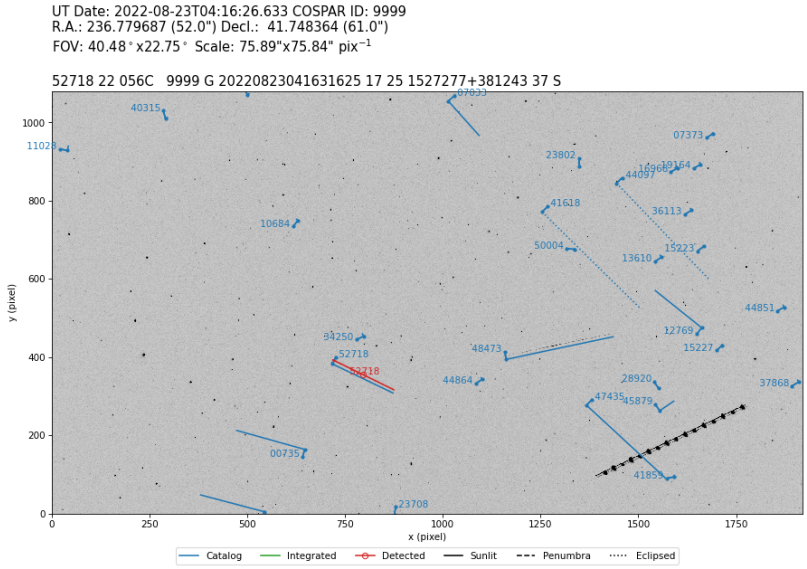



**Figure 10.3:** `stvid` PNG of the satellite trail of NORAD ID 52718 identified in red.

Note, there is no `.dat` file created corresponding to the `2022-08-23T04:16:26.633.fits.png` file, since it doesn't show detected trails.

The `.dat` file contents look like this, for NORAD ID 48473:

```
48473 21 040AX  9999 G 20220823041631625 17 25 1533406+483563 37 S
```

XXX This is the IOD ? XXX The first field is the NORAD ID. The fourth field looks like COSPAR. XXX Large number starting with 2022 is date and time stamp.

The `.dat` file contents look like this, for NORAD ID 52718:

```
52718 22 056C   9999 G 20220823041631625 17 25 1527277+381243 37 S
```

The `.dat` file contents look like this, for the unidentified trail (the airplane):

```
90000 22 500A   9999 G 20220823041631625 17 25 1457613+535844 37 S
```

### 10.3.8 `stvid .csv` Files

During processing, `stvid` generates (at least!) three CSV files. For the observation discussed above, staring with figure 10.1, the following CSV files are generated:

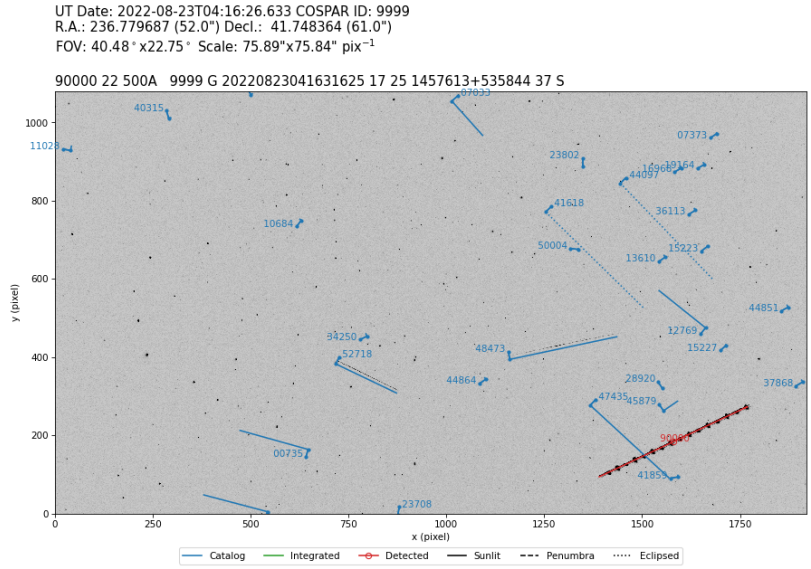



**Figure 10.4:** `stvid` PNG of an unidentified trail in red.

```
2022-08-23T04:16:26.633_hough.csv
2022-08-23T04:16:26.633_predict.csv
2022-08-23T04:16:26.633_threshold.csv
```

### `stvid hough.csv` Files

Full contents of the `2022-08-23T04:16:26.633_hough.csv` file:

```
ax,ay,az,bx,by,bz,n
1585.699766,186.468458,51.073014,0.879795,0.416052,0.229917,3424
794.849123,355.614035,48.596491,0.780147,-0.382158,0.495304,285
1297.660777,432.469965,52.989399,0.923059,0.193205,0.332617,283
```

### `stvid predict.csv` Files

The `2022-08-23T04:16:26.633_predict.csv` is 3,764 lines long. Below is an extract of the lines. I shortened the path, which included the full path to my home directory, for clarity:

```
satno,cospar,mjd,ra,dec,state,tlefile,age
00134,61015U  ,59814.17808603,182.367480,+37.582914,sunlit,/data/tle/catalog.tle,0.441
00134,61015U  ,59814.17809887,182.231429,+37.736730,sunlit,/data/tle/catalog.tle,0.441
00134,61015U  ,59814.17811171,182.094454,+37.890457,sunlit,/data/tle/catalog.tle,0.441
00134,61015U  ,59814.17812455,181.956550,+38.044091,sunlit,/data/tle/catalog.tle,0.441
...
00735,64002C  ,59814.17808603,227.004270,+34.763570,sunlit,/data/tle/catalog.tle,0.027
```

```
00735,64002C  ,59814.17809887,227.192764,+34.379114,sunlit,/data/tle/catalog.tle,0.027
00735,64002C  ,59814.17811171,227.378969,+33.994697,sunlit,/data/tle/catalog.tle,0.027
...
46083,20055BJ ,59814.17808603,204.492490,+40.972573,sunlit,/data/tle/catalog.tle,0.022
46083,20055BJ ,59814.17809887,204.840813,+41.039396,sunlit,/data/tle/catalog.tle,0.022
46083,20055BJ ,59814.17811171,205.193715,+41.105756,sunlit,/data/tle/catalog.tle,0.022
...
32706,08010A  ,59814.17808603,169.555855,+50.073199,sunlit,/data/tle/inttles.tle,0.624
32706,08010A  ,59814.17809887,169.560127,+50.075268,sunlit,/data/tle/inttles.tle,0.624
32706,08010A  ,59814.17811171,169.564400,+50.077336,sunlit,/data/tle/inttles.tle,0.624
...
```

### `stvid threshold.csv` Files

`stvid` creates a `threshold.csv` file with a filename like:

`2022-08-23T04:16:26.633_threshold.csv`

The above file has 5,293 lines. Here are a few of them:

```
58.000000,0.000000,38.000000
69.000000,1.000000,59.000000
1832.000000,1.000000,52.000000
418.000000,2.000000,11.000000
1183.000000,2.000000,71.000000
...
1276.000000,1078.000000,1.000000
11.000000,1079.000000,38.000000
1500.000000,1079.000000,45.000000
```

# Support

# Help!

## 11.1 Support

How to get help.

- `#satnogs-optical` channel in Matrix — `#satnogs-optical:matrix.org`
- LSF Community Forum — `https://community.libre.space`

## Contact

## Email, Chat, Forum

## 12.1 Contact

Jeff Moe

Username: jebba
Email: moe@spacecruft.org

# List of Abbreviations

**V** Volt.

# Acronyms

**API** application programming interface. (application programming interface)

**BDS** BeiDou Navigation Satellite System. (BeiDou Navigation Satellite System)

**BIOS** Basic Input/Output System. (Basic Input/Output System)

**BSD** Berkeley Software Distribution. (Berkeley Software Distribution)

**CCD** charge-coupled device. (charge-coupled device)

**CMOS** Complementary metal-oxide-semiconductor (pronounced "see-moss"). (Complementary metal–oxide–semiconductor)

**COTS** Commodity off the shelf. (Commodity off the shelf)

**CPU** central processing unit. (central processing unit)

**CSV** Comma Separated Value. (Comma Separated Value)

**DC** Direct current. (Direct current)

**DCS** distributed control system. (distributed control system)

**DFSG** Debian Free Software Guidelines is a set of guidelines that the Debian Project uses to determine whether a software license is a Free Software license, which in turn is used to determine whether a piece of software can be included in Debian. The DFSG is part of the Debian Social Contract.[1] (Debian Free Software Guidelines)

**DSLR** digital single-lens reflex. (digital single-lens reflex)

**EQ** Equatorial. (equatorial)

**FITS** Flexible Image Transport System is an open standard defining a digital file format useful for storage, transmission and processing of data: formatted as multi-dimensional arrays (for example a 2D image), or tables. FITS is the most commonly used digital file format in astronomy. The FITS standard was designed specifically for astronomical data, and

[1] Wikipedia contributors. *Debian Free Software Guidelines — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Debian_Free_Software_Guidelines&oldid=1086423235. [Online; accessed 31-August-2022]. 2022

includes provisions such as describing photometric and spatial calibration information, together with image origin metadata.[2] (Flexible Image Transport System)

**FLIR** Forward-looking infrared. (Forward-looking infrared)

**FLOSS** Free/libre and open-source software. See also: FOSS. (Free/libre and open-source software)

**FOSH** Free open-source hardware. See also: OSH. (Free open-source hardware)

**FOSS** free and Open Source software is a term used to refer to groups of software consisting of both Free Software and OSS where anyone is freely licensed to use, copy, study, and change the software in any way, and the source code is openly shared so that people are encouraged to voluntarily improve the design of the software. This is in contrast to proprietary software, where the software is under restrictive copyright licensing and the source code is usually hidden from the users. FOSS maintains the software user's civil liberty rights. Other benefits of using FOSS can include decreased software costs, increased security and stability (especially in regard to malware), protecting privacy, education, and giving users more control over their own hardware. Free and open-source operating systems such as Linux and descendants of BSD are widely utilized today, powering millions of servers, desktops, smartphones (e.g., Android), and other devices. Free-software licenses and Open Source licenses are used by many software packages. The free software movement and the Open Source software movement are online social movements behind widespread production and adoption of FOSS, with the former preferring to use the terms FLOSS or free/libre.[3] (free and open-source software)

**FoV** Field of view. (Field of View)

**GHz** gigahertz. (gigahertz)

**GLONASS** Global Navigation Satellite System. (Global Navigation Satellite System)

**GNSS** Global Navigation Satellite System. (Global Navigation Satellite System)

**GNU** "GNU's Not Unix!" is an extensive collection of Free Software, which can be used as an OS or can be used in parts with other OSs. The use of the completed GNU tools led to the family of operating systems popularly known as Linux. Most of GNU is licensed under the GNU Project's own GPL. GNU is also the project within which the Free Software concept originated.[4] (GNU's Not Unix!)

**GPL** GNU General Public License is a series of widely used Free Software licenses that guarantee end users the four freedoms to run, study, share, and modify the software. The license was the first copyleft for general use. Historically, the GPL license family has been

[2] Wikipedia contributors, *FITS — Wikipedia, The Free Encyclopedia*

[3] Wikipedia contributors. *Free and open-source software — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Free_and_open-source_software&oldid=1106343648. [Online; accessed 30-August-2022]. 2022

[4] Wikipedia contributors. *GNU — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=GNU&oldid=1103488294. [Online; accessed 28-August-2022]. 2022

one of the most popular software licenses in the FLOSS software domain.[5] (GNU General Public License)

**GPS** Global Positioning System. (Global Positioning System)

**HDL** Hardware description language. (Hardware description language)

**Hz** hertz. (hertz)

**INDI** Instrument Neutral Distributed Interface is a DCS protocol to enable control, data acquisition and exchange among hardware devices and software front ends, emphasizing astronomical instrumentation.[6] (Instrument Neutral Distributed Interface)

**IP** Internet Protocol Address. (Internet Protocol Address)

**IP67** Ingress Protection code 67. (Ingress Protection code 67)

**IPv6** Internet Protocol version 6. (Internet Protocol version 6)

**ISS** International Space Station. (International Space Station)

**ITU** The International Telecommunication Union is a specialized agency of the United Nations responsible for many matters related to information and communication technologies.[7] (International Telecommunication Union)

**kHz** kilohertz. (kilohertz)

**LSF** Libre Space Foundation is a non-profit foundation registered since 2015 in Greece and the creators of the SatNOGS project.[8] (Libre Space Foundation)

**MHz** megahertz. (megahertz)

**MIPI** Mobile Industry Processor Interface. (Mobile Industry Processor Interface)

**NASA** is an independent agency of the US federal government responsible for the civil space program, aeronautics research, and space research.[9] (National Aeronautics and Space Administration)

**NGC** New General Catalogue of Nebulae and Clusters of Stars. (New General Catalogue of Nebulae and Clusters of Stars)

---

[5] Wikipedia contributors. *GNU General Public License — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=GNU_General_Public_License&oldid=1106596497. [Online; accessed 28-August-2022]. 2022

[6] Wikipedia contributors. *Instrument Neutral Distributed Interface — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Instrument_Neutral_Distributed_Interface&oldid=1062506145. [Online; accessed 31-August-2022]. 2021

[7] Wikipedia contributors. *International Telecommunication Union — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=International_Telecommunication_Union&oldid=1105915792. [Online; accessed 28-August-2022]. 2022

[8] https://libre.space/about-us/

[9] Wikipedia contributors. *NASA — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=NASA&oldid=1107567899. [Online; accessed 30-August-2022]. 2022

**NORAD** North American Aerospace Defense Command is a combined organization of the United States and Canada that provides aerospace warning, air sovereignty, and protection for Canada and the continental United States.[10] (North American Aerospace Defense Command)

**NORAD ID** is a sequential nine-digit number assigned by the USSPACECOM in the order of launch or discovery to all artificial objects in the orbits of Earth and those that left Earth's orbits. Space Command shares the catalog via `space-track.org`, which is maintained by the 18th SDS. NORAD ID is also known as a satellite catalog number.[11] (North American Aerospace Defense Catalog Number)

**NTP** Network Time Protocol. (Network Time Protocol)

**NUC** Next Unit of Computing. (Next Unit of Computing)

**OS** An operating system is system software that manages computer hardware, software resources, and provides common services for computer programs.[12] (Operating System)

**OSAT** open-source appropriate technology. (open-source appropriate technology)

**OSH** open-source hardware consists of physical artifacts of technology designed and offered by the open-design movement. Both FOSS and open-source hardware are created by the open-source culture movement and apply a like concept to a variety of components. It is sometimes, thus, referred to as FOSH. The term usually means that information about the hardware is easily discerned so that others can make it—coupling it closely to the maker movement. Hardware design (i.e. mechanical drawings, schematics, bills of material, PCB layout data, HDL source code and integrated circuit layout data), in addition to the software that drives the hardware, are all released under free/libre terms. The original sharer gains feedback and potentially improvements on the design from the FOSH community. There is now significant evidence that such sharing can drive a high return on investment for the scientific community. It is not enough to merely use an Open Source license; an Open Source product or project will follow Open Source principles, such as modular design and community collaboration.[13] (open-source hardware)

**OSS** open-source software is computer software that is released under a license in which the copyright holder grants users the rights to use, study, change, and distribute the software and its source code to anyone and for any purpose. Open-source software may be developed in a collaborative public manner. Open-source software is a prominent example of open collaboration, meaning any capable user is able to participate online in development, making the number of possible contributors indefinite. The ability to examine the code facilitates public trust in the software. Open source code can be used for studying and allows capable end users to adapt software to their personal needs in a similar way user

[10] Wikipedia contributors. *NORAD — Wikipedia, The Free Encyclopedia*. `https://en.wikipedia.org/w/index.php?title=NORAD&oldid=1105457081`. [Online; accessed 30-August-2022]. 2022

[11] Wikipedia contributors. *Satellite Catalog Number — Wikipedia, The Free Encyclopedia*. `https://en.wikipedia.org/w/index.php?title=Satellite_Catalog_Number&oldid=1100318368`. [Online; accessed 4-September-2022]. 2022

[12] Wikipedia contributors. *Operating system — Wikipedia, The Free Encyclopedia*. `https://en.wikipedia.org/w/index.php?title=Operating_system&oldid=1104687084`. [Online; accessed 3-September-2022]. 2022

[13] Wikipedia contributors. *Open-source hardware — Wikipedia, The Free Encyclopedia*. `https://en.wikipedia.org/w/index.php?title=Open-source_hardware&oldid=1105615869`. [Online; accessed 30-August-2022]. 2022

scripts and custom style sheets allow for web sites, and eventually publish the modification as a fork for users with similar preference.[14] (open-source software)

**PaaS** Platform as a Service. (Platform as a Service)

**PC** Personal Computer. (Personal Computer)

**PCB** Printed circuit board. (Printed Circuit Board)

**PIP** Package Installer for Python. (Package Installer for Python)

**PNG** Portable Network Graphics. (Portable Network Graphics)

**PoE** Power over Ethernet. (Power over Ethernet)

**PPS** Pulse per second. (Pulse per second)

**PTZ** Pan-tilt-zoom. (Pan-tilt-zoom)

**RAM** Random-access memory. (Random-access memory)

**RASA** Rowe-Ackermann Schmidt Astrograph. (Rowe-Ackermann Schmidt Astrograph)

**RF** Radio frequency is the oscillation rate of an alternating electric current or voltage or of a magnetic, electric or electromagnetic field or mechanical system in the frequency range from around 20 kHz to around 300 GHz. This is roughly between the upper limit of audio frequencies and the lower limit of infrared frequencies; these are the frequencies at which energy from an oscillating current can radiate off a conductor into space as radio waves.[15] (radio frequency)

**SBC** Single board computer. (Single board computer)

**SDK** Software development kit. (Software development kit)

**SDP** Simplified Deep Space Perturbations models apply to objects with an orbital period greater than 225 minutes, which corresponds to an altitude of 5,877.5 km, assuming a circular orbit. See also: SGP.[16] (Simplified Deep Space Perturbations)

**SDR** Software-defined radio is a radio communication system where components that have been traditionally implemented in analog hardware (e.g. mixers, filters, amplifiers, modulators/demodulators, detectors, etc.) are instead implemented by means of software on a PC or embedded system. SDRs are used in SatNOGS ground stations.[17] (Software-defined radio)

**SDS** Space Defense Squadron. (Space Defense Squadron)

[14] Wikipedia contributors. *Open-source software — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Open-source_software&oldid=1104440334. [Online; accessed 30-August-2022]. 2022

[15] Wikipedia contributors. *Radio frequency — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Radio_frequency&oldid=1104615064. [Online; accessed 28-August-2022]. 2022

[16] Wikipedia contributors. *Simplified perturbations models — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Simplified_perturbations_models&oldid=983925578. [Online; accessed 30-August-2022]. 2020

[17] Wikipedia contributors, *Software-defined radio — Wikipedia, The Free Encyclopedia*

**SGP** Simplified General Perturbations models apply to near earth objects with an orbital period of less than 225 minutes. Simplified perturbations models are a set of five mathematical models (SGP, SGP4, SDP4, SGP8 and SDP8) used to calculate orbital state vectors of satellites and space debris relative to the Earth-centered inertial coordinate system. This set of models is often referred to collectively as SGP4 due to the frequency of use of that model particularly with TLE sets produced by NORAD and NASA. These models predict the effect of perturbations caused by the Earth's shape, drag, radiation, and gravitation effects from other bodies such as the sun and moon. See also: SDP.[18] (Simplified General Perturbations)

**SIDLOC** Spacecraft Identification and Localization. (Spacecraft Identification and Localization)

**SIP** Simple Imaging Polynomial. (Simple Imaging Polynomial)

**SSA** Space Situational Awareness. (Space Situational Awareness)

**TLE** two-line element set is a data format encoding a list of orbital elements of an Earth-orbiting object for a given point in time, the epoch. Using a suitable prediction formula, the state (position and velocity) at any point in the past or future can be estimated to some accuracy. TLEs can describe the trajectories only of Earth-orbiting objects. TLEs are widely used as input for projecting the future orbital tracks of space debris for purposes of characterizing "future debris events to support risk analysis, close approach analysis, collision avoidance maneuvering" and forensic analysis. The format was originally intended for punched cards, encoding a set of elements on two standard 80-column cards.[19] (two-line element set)

**UHF** Ultra High Frequency is the ITU designation for radio frequencies in the range between 300 MHz and 3 GHz, also known as the decimetre band as the wavelengths range from one meter to one tenth of a meter (one decimeter). Lower frequency signals fall into the VHF or lower bands.[20] (Ultra High Frequency)

**USB** Universal Serial Bus. (Universal Serial Bus)

**USM** is a type of piezoelectric motor powered by the ultrasonic vibration of a component, the stator, placed against another component, the rotor or slider depending on the scheme of operation (rotation or linear translation). One common application of ultrasonic motors is in camera lenses where they are used to move lens elements as part of the auto-focus system.[21] (Ultrasonic motor)

**USSPACECOM** United States Space Command. (United States Space Command)

**UVC** USB video device class. (USB video device class)

[18] Wikipedia contributors, *Simplified perturbations models — Wikipedia, The Free Encyclopedia*

[19] Wikipedia contributors. *Two-line element set — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Two-line_element_set&oldid=1104028347. [Online; accessed 28-August-2022]. 2022

[20] Wikipedia contributors. *Ultra high frequency — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Ultra_high_frequency&oldid=1096417717. [Online; accessed 28-August-2022]. 2022

[21] Wikipedia contributors. *Ultrasonic motor — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Ultrasonic_motor&oldid=1099892445. [Online; accessed 4-September-2022]. 2022

**V4L2** Video for Linux v2. (Video for Linux v2)

**VHF** Very High Frequency is the ITU designation for the range of radio frequency electromagnetic waves (radio waves) from 30 to 300 MHz, with corresponding wavelengths of ten meters to one meter. Frequencies immediately below VHF are denoted high frequency (HF), and the next higher frequencies are known as UHF.[22] (Very High Frequency)

**WCS** World Coordinate System. (World Coordinate System)

[22] Wikipedia contributors. *Very high frequency — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Very_high_frequency&oldid=1105564543. [Online; accessed 28-August-2022]. 2022

# Glossary

**altazimuth mount** a simple two-axis mount for supporting and rotating an instrument about two perpendicular axes — one vertical and the other horizontal. Rotation about the vertical axis varies the azimuth (compass bearing) of the pointing direction of the instrument. Rotation about the horizontal axis varies the altitude angle (angle of elevation) of the pointing direction. These mounts are used, for example, with telescopes, cameras, and radio antennas.[23]

**amateur radio** is the use of the radio frequency spectrum for purposes of non-commercial exchange of messages, wireless experimentation, self-training, private recreation, radiosport, contesting, and emergency communications.[24]

**antenna** the interface between radio waves propagating through space and electric currents moving in metal conductors, used with a transmitter or receiver.[25]

**appropriate technology** is a movement (and its manifestations) encompassing technological choice and application that is small-scale, affordable by locals, decentralized, labor-intensive, energy-efficient, environmentally sustainable, and locally autonomous. Appropriate technology has been used to address issues in a wide range of fields. Today appropriate technology is often developed using Open Source principles, which have led to OSAT and thus many of the plans of the technology can be freely found on the Internet.[26]

**astrograph** is a telescope designed for the sole purpose of astrophotography. Astrographs are mostly used in wide-field astronomical surveys of the sky and for detection of objects such as asteroids, meteors, and comets.[27]

**AstroImageJ** Application for astronomy and satellite image analysis.[28]

**astronomical catalogue** is a list or tabulation of astronomical objects, typically grouped together because they share a common type, morphology, origin, means of detection, or method of discovery. The oldest and largest are star catalogues. Hundreds have been published,

[23] Wikipedia contributors. *Altazimuth mount — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Altazimuth_mount&oldid=1056074953. [Online; accessed 28-August-2022]. 2021

[24] Wikipedia contributors. *Amateur radio — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Amateur_radio&oldid=1106068703. [Online; accessed 2-September-2022]. 2022

[25] Wikipedia contributors. *Antenna (radio) — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Antenna_(radio)&oldid=1104603350. [Online; accessed 28-August-2022]. 2022

[26] Wikipedia contributors. *Appropriate technology — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Appropriate_technology&oldid=1104787534. [Online; accessed 2-September-2022]. 2022

[27] Wikipedia contributors. *Astrograph — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Astrograph&oldid=1106095357. [Online; accessed 30-August-2022]. 2022

[28] https://www.astro.louisville.edu/software/astroimagej/

including general ones and special ones for such items as infrared stars, variable stars, giant stars, multiple star systems, and star clusters. Since the late 20th century catalogs are increasingly often compiled by computers from an automated survey, and published as computer files rather than on paper.[29]

**binning** is the process of combining adjacent pixels throughout an image, by summing or averaging their values, during or after readout. Charge from adjacent pixels in CCD image sensors and some other image sensors can be combined during readout, increasing the line rate or frame rate. In the context of image processing, binning is the procedure of combining clusters of adjacent pixels, throughout an image, into single pixels. For example, in 2x2 binning, an array of 4 pixels becomes a single larger pixel, reducing the number of pixels to 1/4 and halving the image resolution in each dimension. The result can be the sum, average, median, minimum, or maximum value of the cluster. This aggregation, although associated with loss of information, reduces the amount of data to be processed, facilitating analysis. The binned image has lower resolution, but the relative noise level in each pixel is generally reduced. Also called pixel binning.[30]

**C** is a general-purpose computer programming language. It was created in the 1970s by Dennis Ritchie, and remains very widely used and influential. By design, C's features cleanly reflect the capabilities of the targeted CPUs. It has found lasting use in OSs, device drivers, protocol stacks, though decreasingly for application software. C is commonly used on computer architectures that range from the largest supercomputers to the smallest microcontrollers and embedded systems. C is used in the sattools suite of applications.[31]

**Celestron** is an American company based in Torrance, California, United States, that manufactures telescopes and distributes telescopes, binoculars, spotting scopes, microscopes, and accessories.[32]

**CubeSat** is a class of miniaturized satellite based around a form factor consisting of 10 cm (3.9 in) cubes. CubeSats have a mass of no more than 2 kg per unit, and often use COTS components for their electronics and structure. CubeSats are put into orbit by deployers on the ISS, or launched as secondary payloads on a launch vehicle. More than a thousand CubeSats have been launched.[33]

**daemon** a service in a Unix OS.[34]

**Debian** a GNU/Linux distribution composed of free and open-source software, developed by the community-supported Debian Project, which was established by Ian Murdock on August 16, 1993. Debian is the basis for many other distributions, notably Ubuntu. Debian is one of the oldest operating systems based on the Linux kernel.[35]

---

[29] Wikipedia contributors. *Astronomical catalog — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Astronomical_catalog&oldid=1035999452. [Online; accessed 2-September-2022]. 2021

[30] Wikipedia contributors. *Pixel binning — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Pixel_binning&oldid=1085651660. [Online; accessed 4-September-2022]. 2022

[31] Wikipedia contributors. *C (programming language) — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=C_(programming_language)&oldid=1106686212. [Online; accessed 30-August-2022]. 2022

[32] Wikipedia contributors. *Celestron — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Celestron&oldid=1091418493. [Online; accessed 2-September-2022]. 2022

[33] Wikipedia contributors. *CubeSat — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=CubeSat&oldid=1106408835. [Online; accessed 30-August-2022]. 2022

[34] Wikipedia contributors. *Daimon (disambiguation) — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Daimon_(disambiguation)&oldid=1069803376. [Online; accessed 3-September-2022]. 2022

[35] Wikipedia contributors. *Debian — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Debian&oldid=1105900486. [Online; accessed 28-August-2022]. 2022

**Docker** is a set of PaaS products that use OS-level virtualization to deliver software in packages called containers.[36]

**embedded system** is a computer system—a combination of a computer processor, computer memory, and input/output peripheral devices—that has a dedicated function within a larger mechanical or electronic system. It is embedded as part of a complete device often including electrical or electronic hardware and mechanical parts. Because an embedded system typically controls physical operations of the machine that it is embedded within, it often has real-time computing constraints. embedded systems control many devices in common use today. it was estimated that ninety-eight percent of all microprocessors manufactured were used in embedded systems.[37]

**firmware** firmware is a specific class of computer software that provides the low-level control for a device's specific hardware. Firmware, such as the BIOS of a PC, may contain basic functions of a device, and may provide hardware abstraction services to higher-level software such as OSs. For less complex devices, firmware may act as the device's complete OS, performing all control, monitoring and data manipulation functions. Typical examples of devices containing firmware are embedded systems (running embedded software), home and personal-use appliances, computers, and computer peripherals.[38]

**Free Software** or libre software, is computer software distributed under terms that allow users to run the software for any purpose as well as to study, change, and distribute it and any adapted versions. Free software is a matter of liberty, not price; all users are legally free to do what they want with their copies of a free software (including profiting from them) regardless of how much is paid to obtain the program. Computer programs are deemed "free" if they give end-users (not just the developer) ultimate control over the software and, subsequently, over their devices.[39]

**GoTo** In amateur astronomy, "GoTo" refers to a type of telescope mount and related software that can automatically point a telescope at astronomical objects that the user selects. Both axes of a GoTo mount are driven by a motor and controlled by a computer. It may be either a microprocessor-based integrated controller or an external PC. This differs from the single-axis semi-automated tracking of a traditional clock-drive equatorial mount. The user can command the mount to point the telescope to the celestial coordinates that the user inputs, or to objects in a pre-programmed database including ones from the Messier catalogue, the NGC, and even major Solar System bodies (the Sun, Moon, and planets). Like a standard EQ mount, EQ GoTo mounts can track the night sky by driving the right ascension axis. Since both axes are computer controlled, GoTo technology also allows telescope manufacturers to add EQ tracking to mechanically simpler altazimuth mounts.[40]

---

[36] Wikipedia contributors. *Docker (software) — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Docker_(software)&oldid=1108127182. [Online; accessed 4-September-2022]. 2022

[37] Wikipedia contributors. *Embedded system — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Embedded_system&oldid=1106874216. [Online; accessed 2-September-2022]. 2022

[38] Wikipedia contributors. *Firmware — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Firmware&oldid=1099439511. [Online; accessed 2-September-2022]. 2022

[39] Wikipedia contributors. *Free software — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Free_software&oldid=1106538260. [Online; accessed 28-August-2022]. 2022

[40] Wikipedia contributors. *GoTo (telescopes) — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=GoTo_(telescopes)&oldid=841595201. [Online; accessed 2-September-2022]. 2018

**gPhoto** is a set of software applications and libraries for use in digital photography. gPhoto supports not just retrieving of images from camera devices, but also upload and remote controlled configuration and capture, depending on whether the camera supports those features. gPhoto supports more than 2500 cameras.[41]

**gpsd** is a computer software program that collects data from a GPS receiver and provides the data via a network to potentially multiple client applications in a server-client application architecture. Gpsd may be run as a daemon to operate transparently as a background task of the server. The network interface provides a standardized data format for multiple concurrent client applications.[42]

**Grafana** is a multi-platform Open Source analytics and interactive visualization web application. It provides charts, graphs, and alerts for the web when connected to supported data sources. Users can create complex monitoring dashboards using interactive query builders. It is used by the SatNOGS project to visualize satellite telemetry.[43]

**ground station** a setup of equipment such as computers, cameras, SDRs, antennas, and receivers, located on Earth, observing space.

**hough3d-code** is a software application for Iterative Hough Transform for Line Detection in 3D Point Clouds.[44]

**KStars** is a planetarium program. It provides an accurate graphical representation of the night sky, from any location on Earth, at any date and time. The display includes up to 100 million stars (with additional addons), 13,000 deep sky objects, constellations from different cultures, all 8 planets, the Sun and Moon, and thousands of comets, asteroids, satellites, and supernovae. It has features to appeal to users of all levels, from informative hypertext articles about astronomy, to robust control of telescopes and CCD cameras, and logging of observations of specific objects.[45]

**Lagrange point** are points of equilibrium for small-mass objects under the influence of two massive orbiting bodies. At the Lagrange points, the gravitational forces of the two large bodies and the centrifugal force balance each other. This can make Lagrange points an excellent location for satellites, as few orbit corrections are needed to maintain the desired orbit. Small objects placed in orbit at Lagrange points are in equilibrium in at least two directions relative to the center of mass of the large bodies.[46]

**libre** The English adjective free is commonly used in one of two meanings: "at no monetary cost" (gratis) and "with little or no restriction" (libre). This ambiguity of free can cause issues where the distinction is important, as it often is in dealing with laws concerning the use of information, such as copyright and patents. The terms gratis and libre may be used to categorise computer programs, according to the licenses and legal restrictions that cover them, in the Free Software and Open Source communities, as well as the broader free culture movement. For example, they are used to distinguish freeware

[41] Wikipedia contributors. *GPhoto — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=GPhoto&oldid=1097546275. [Online; accessed 2-September-2022]. 2022

[42] Wikipedia contributors. *Gpsd — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Gpsd&oldid=1076207218. [Online; accessed 2-September-2022]. 2022

[43] Wikipedia contributors. *Grafana — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Grafana&oldid=1104046322. [Online; accessed 30-August-2022]. 2022

[44] https://gitlab.com/pierros/hough3d-code

[45] Wikipedia contributors. *KStars — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=KStars&oldid=1076667045. [Online; accessed 2-September-2022]. 2022

[46] Wikipedia contributors. *Lagrange point — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Lagrange_point&oldid=1102894784. [Online; accessed 2-September-2022]. 2022

(software gratis) from Free Software (software libre). "Think free as in free speech, not free beer." – Richard Stallman.[47]

**Linux** is a free and open-source, monolithic, modular, multitasking, Unix-like operating system kernel. It was originally authored in 1991 by Linus Torvalds for his i386-based PC, and it was soon adopted as the kernel for the GNU OS, which was written to be a free (libre) replacement for Unix.[48]

**mast** typically tall structures designed to support antennas for telecommunications and broadcasting.[49]

**Matrix** an open standard and communication protocol for real-time communication.[50]

**Open Source** is source code that is made freely available for possible modification and redistribution. Products include permission to use the source code, design documents, or content of the product. The open-source model is a decentralized software development model that encourages open collaboration. A main principle of open-source software development is peer production, with products such as source code, blueprints, and documentation freely available to the public. The open-source movement in software began as a response to the limitations of proprietary code. The model is used for projects such as in open-source appropriate technology.[51]

**OpenCV** Open Source Computer Vision Library is a library of programming functions mainly aimed at real-time computer vision.[52]

**optical ground station** a ground station using optical equipment (cameras) instead of antennas.

**orbit** is the curved trajectory of an object such as the trajectory of a planet around a star, or of a natural satellite around a planet, or of an artificial satellite around an object or position in space such as a planet, moon, asteroid, or Lagrange point. Normally, orbit refers to a regularly repeating trajectory, although it may also refer to a non-repeating trajectory. To a close approximation, planets and satellites follow elliptic orbits, with the center of mass being orbited at a focal point of the ellipse, as described by Kepler's laws of planetary motion. For most situations, orbital motion is adequately approximated by Newtonian mechanics, which explains gravity as a force obeying an inverse-square law. However, Albert Einstein's general theory of relativity, which accounts for gravity as due to curvature of spacetime, with orbits following geodesics, provides a more accurate calculation and understanding of the exact mechanics of orbital motion.[53]

**perturbation** is the complex motion of a massive body subjected to forces other than the gravitational attraction of a single other massive body. The other forces can include

[47] Wikipedia contributors. *Gratis versus libre — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Gratis_versus_libre&oldid=1085552810. [Online; accessed 30-August-2022]. 2022

[48] Wikipedia contributors. *Linux kernel — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Linux_kernel&oldid=1105840074. [Online; accessed 28-August-2022]. 2022

[49] Wikipedia contributors. *Radio masts and towers — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Radio_masts_and_towers&oldid=1103964392. [Online; accessed 28-August-2022]. 2022

[50] Wikipedia contributors. *Matrix (protocol) — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Matrix_(protocol)&oldid=1106244486. [Online; accessed 28-August-2022]. 2022

[51] Wikipedia contributors. *Open source — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Open_source&oldid=1103126225. [Online; accessed 28-August-2022]. 2022

[52] Wikipedia contributors. *OpenCV — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=OpenCV&oldid=1106133881. [Online; accessed 2-September-2022]. 2022

[53] Wikipedia contributors. *Orbit — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Orbit&oldid=1106406646. [Online; accessed 30-August-2022]. 2022

a third (fourth, fifth, etc.) body, resistance, as from an atmosphere, and the off-center attraction of an oblate or otherwise misshapen body.[54]

**photon** is an elementary particle that is a quantum of the electromagnetic field, including electromagnetic radiation such as light and radio waves, and the force carrier for the electromagnetic force. Photons are massless, so they always move at the speed of light in vacuum, 299,792,458 meters/second.[55]

**pipeline** is a set of data processing elements connected in series, where the output of one element is the input of the next one. The elements of a pipeline are often executed in parallel or in time-sliced fashion.[56]

**plate solver** is software implementing a technique used in astronomy and applied on celestial images. Solving an image is finding match between the imaged stars and a star catalogue. The solution is a math model describing the corresponding astronomical position of each image pixel. The position of reference catalogue stars has to be known to a high accuracy so an astrometric reference catalogue is used. The image solution contains a reference point, often the image centre, image scale, image orientation and in some cases an image distortion model. With the astrometric solution it is possible to: 1) Calculate the celestial coordinates of any object on the image. 2) Synchronize the telescope mount or satellite pointing position to the center of the image taken. Astrometric solving programs extract the star x,y positions from the celestial image, groups them in three-star triangles or four-star quads. Then it calculates for each group a geometric hash code based on the distance and/or angles between the stars in the group. It then compares the resulting hash codes with the hash codes created from catalogue stars to find a match. If it finds sufficient statistically reliable matches, it can calculate transformation factors. There are several conventions to model the transformation from image pixel location to the corresponding celestial coordinates. The simplest linear model is called the WCS. A more advanced convention is SIP describing the transformation in polynomials to cope with non-linear geometric distortion in the celestial image, mainly caused by the optics.[57]

**Python** is a high-level, interpreted, general-purpose programming language. Its design philosophy emphasizes code readability. It is often described as a "batteries included" language due to its comprehensive standard library. Python consistently ranks as one of the most popular programming languages. It is one of the main languages of the SatNOGS project and stvid.[58]

**Raspberry Pi** is a series of small SBCs. It is typically used by computer and electronic hobbyists as an embedded system.[59]

---

[54] Wikipedia contributors. *Perturbation (astronomy) — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Perturbation_(astronomy)&oldid=1105431363. [Online; accessed 30-August-2022]. 2022

[55] Wikipedia contributors. *Photon — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Photon&oldid=1104709214. [Online; accessed 2-September-2022]. 2022

[56] Wikipedia contributors. *Pipeline (computing) — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Pipeline_(computing)&oldid=1096885455. [Online; accessed 2-September-2022]. 2022

[57] Wikipedia contributors. *Astrometric solving — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Astrometric_solving&oldid=1099832612. [Online; accessed 30-August-2022]. 2022

[58] Wikipedia contributors. *Python (programming language) — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Python_(programming_language)&oldid=1107007073. [Online; accessed 30-August-2022]. 2022

[59] Wikipedia contributors. *Raspberry Pi — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Raspberry_Pi&oldid=1107949674. [Online; accessed 2-September-2022]. 2022

**rotator** a device used to change the orientation, within the horizontal plane, of a directional antenna. Most antenna rotators have two parts, the rotator unit and the controller. The controller is normally placed near the equipment which the antenna is connected to, while the rotator is mounted on the antenna mast directly below the antenna. Rotators are commonly used in amateur radio.[60]

**satellite** is an object intentionally placed into orbit in outer space. Except for passive satellites, most satellites have an electricity generation system for equipment on board. Most satellites also have a method of communication to ground stations, called transponders. Many satellites use a standardized bus to save cost and work, the most popular of which is small CubeSats. Similar satellites can work together as a group, forming constellations.[61]

**satid** is part of sattools.[62]

**SatNOGS** Open Source global network of satellite ground stations.[63]

**SatNOGS DB** is an effort to create an hollistic, unified, global database for all artificial objects in space (satellites and spacecrafts). Users can view and export the data, contribute to it, or connect applications using an API. It is part of the SatNOGS project.[64]

**SatNOGS Optical** is a project by the LSF to add optical ground stations to the SatNOGS network.[65]

**satpredict** is a software application to compute satellite predictions. It is used by stvid.[66]

**sattools** Satellite Tracking Toolkit. The main `sattools` applications are being ported to stvid and other related Python applications.[67]

**sky chart** or star chart or star map, also called or sky map, is a map of the night sky. Astronomers divide these into grids to use them more easily. They are used to identify and locate constellations and astronomical objects such as stars, nebulae, and galaxies. They have been used for human navigation since time immemorial. Note that a sky chart differs from an astronomical catalogue, which is a listing or tabulation of astronomical objects for a particular purpose.[68]

**skymap** is part of sattools. Visualize satellites on a map of the sky.[69]

**slew** The process of rotating a telescope to observe a different region of the sky.[70]

**software repository** or repo for short, is a storage location for software packages. Often a table of contents is also stored, along with metadata. A software repository is

[60] Wikipedia contributors. *Antenna rotator — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Antenna_rotator&oldid=1064620974. [Online; accessed 28-August-2022]. 2022

[61] Wikipedia contributors. *Satellite — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Satellite&oldid=1106286374. [Online; accessed 30-August-2022]. 2022

[62] https://github.com/cbassa/sattools/blob/master/satid.c

[63] https://satnogs.org/

[64] https://db.satnogs.org/

[65] https://satnogs.org/

[66] https://github.com/cbassa/satpredict

[67] https://github.com/cbassa/sattools

[68] Wikipedia contributors. *Star chart — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Star_chart&oldid=1106250278. [Online; accessed 2-September-2022]. 2022

[69] https://github.com/cbassa/sattools/blob/master/skymap.c

[70] Wikipedia contributors. *Slewing — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Slewing&oldid=1061997651. [Online; accessed 2-September-2022]. 2021

typically managed by source control or repository managers. Package managers allow automatically installing and updating repositories (sometimes called "packages").[71]

**star catalogue** is an astronomical catalogue that lists stars. In astronomy, many stars are referred to simply by catalogue numbers. There are a great many different star catalogues which have been produced for different purposes over the years. Most modern catalogues are available in electronic format and can be freely downloaded from space agencies' data centres. The largest is being compiled from the spacecraft Gaia and thus far has over a billion stars. Completeness and accuracy are described by the faintest limiting magnitude and the accuracy of the positions.[72]

**star trail** is a type of photograph that uses long exposure times to capture diurnal circles, the apparent motion of stars in the night sky due to Earth's rotation. A star-trail photograph shows individual stars as streaks across the image, with longer exposures yielding longer arcs.[73]

**stvid** Satellite tools video application for acquiring and processing sky images.[74]

**Teledyne FLIR** a subsidiary of Teledyne Technologies, specializes in the design and production of thermal imaging cameras and sensors. The name is based on the acronym FLIR.[75]

**telemetry** is the in situ collection of measurements or other data at remote points and their automatic transmission to receiving equipment (telecommunication) for monitoring. The word is derived from the Greek roots tele, "remote", and metron, "measure". Although the term commonly refers to wireless data transfer mechanisms (e.g., using radio, ultrasonic, or infrared systems), it also encompasses data transferred over other media such as a telephone or computer network, optical link or other wired communications like power line carriers.[76]

**telescope** is an optical instrument using lenses, curved mirrors, or a combination of both to observe distant objects, or various devices used to observe distant objects by their emission, absorption, or reflection of electromagnetic radiation. The word telescope now refers to a wide range of instruments capable of detecting different regions of the electromagnetic spectrum, and in some cases other types of detectors.[77]

**toolchain** is a set of programming tools that is used to perform a complex software development task or to create a software product, which is typically another computer program or a set of related programs.[78]

---

[71] Wikipedia contributors. *Software repository — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Software_repository&oldid=1081326130. [Online; accessed 2-September-2022]. 2022

[72] Wikipedia contributors. *Star catalogue — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Star_catalogue&oldid=1107326057. [Online; accessed 2-September-2022]. 2022

[73] Wikipedia contributors. *Star trail — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Star_trail&oldid=1075094531. [Online; accessed 2-September-2022]. 2022

[74] https://github.com/cbassa/stvid

[75] Wikipedia contributors. *Teledyne FLIR — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Teledyne_FLIR&oldid=1102704879. [Online; accessed 3-September-2022]. 2022

[76] Wikipedia contributors. *Telemetry — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Telemetry&oldid=1080706265. [Online; accessed 30-August-2022]. 2022

[77] Wikipedia contributors. *Telescope — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Telescope&oldid=1106167988. [Online; accessed 30-August-2022]. 2022

[78] Wikipedia contributors. *Toolchain — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Toolchain&oldid=1094527160. [Online; accessed 2-September-2022]. 2022

**Unix** is a family of multitasking, multiuser computer OSs that derive from the original AT&T Unix, whose development started in 1969 at the Bell Labs research center by Ken Thompson, Dennis Ritchie, and others.[79]

**upstream** refers to a direction toward the original authors or maintainers of software that is distributed as source code, and is a qualification of either a version (released by the original authors, based on their upstream source code), a bug or a patch.[80]

[79] Wikipedia contributors. *Unix — Wikipedia, The Free Encyclopedia.* `https://en.wikipedia.org/w/index.php?title=Unix&oldid=1105826457`. [Online; accessed 2-September-2022]. 2022

[80] Wikipedia contributors. *Upstream (software development) — Wikipedia, The Free Encyclopedia*. `https://en.wikipedia.org/w/index.php?title=Upstream_(software_development)&oldid=1066370459`. [Online; accessed 3-September-2022]. 2022

# Bibliography


Wikipedia contributors. *Altazimuth mount — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Altazimuth_mount&oldid=1056074953. [Online; accessed 28-August-2022]. 2021.

— *Amateur radio — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Amateur_radio&oldid=1106068703. [Online; accessed 2-September-2022]. 2022.

— *Antenna (radio) — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Antenna_(radio)&oldid=1104603350. [Online; accessed 28-August-2022]. 2022.

— *Antenna rotator — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Antenna_rotator&oldid=1064620974. [Online; accessed 28-August-2022]. 2022.

— *Appropriate technology — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Appropriate_technology&oldid=1104787534. [Online; accessed 2-September-2022]. 2022.

— *Astrograph — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Astrograph&oldid=1106095357. [Online; accessed 30-August-2022]. 2022.

— *Astrometric solving — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Astrometric_solving&oldid=1099832612. [Online; accessed 30-August-2022]. 2022.

— *Astronomical catalog — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Astronomical_catalog&oldid=1035999452. [Online; accessed 2-September-2022]. 2021.

— *C (programming language) — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=C_(programming_language)&oldid=1106686212. [Online; accessed 30-August-2022]. 2022.

— *Celestron — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Celestron&oldid=1091418493. [Online; accessed 2-September-2022]. 2022.

— *CubeSat — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=CubeSat&oldid=1106408835. [Online; accessed 30-August-2022]. 2022.

Wikipedia contributors. *Daimon (disambiguation) — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Daimon_(disambiguation)&oldid=1069803376. [Online; accessed 3-September-2022]. 2022.
— *Debian — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Debian&oldid=1105900486. [Online; accessed 28-August-2022]. 2022.
— *Debian Free Software Guidelines — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Debian_Free_Software_Guidelines&oldid=1086423235. [Online; accessed 31-August-2022]. 2022.
— *Docker (software) — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Docker_(software)&oldid=1108127182. [Online; accessed 4-September-2022]. 2022.
— *Embedded system — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Embedded_system&oldid=1106874216. [Online; accessed 2-September-2022]. 2022.
— *Firmware — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Firmware&oldid=1099439511. [Online; accessed 2-September-2022]. 2022.
— *FITS — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=FITS&oldid=1091431488. [Online; accessed 28-August-2022]. 2022.
— *Free and open-source software — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Free_and_open-source_software&oldid=1106343648. [Online; accessed 30-August-2022]. 2022.
— *Free software — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Free_software&oldid=1106538260. [Online; accessed 28-August-2022]. 2022.
— *GNU — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=GNU&oldid=1103488294. [Online; accessed 28-August-2022]. 2022.
— *GNU General Public License — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=GNU_General_Public_License&oldid=1106596497. [Online; accessed 28-August-2022]. 2022.
— *GoTo (telescopes) — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=GoTo_(telescopes)&oldid=841595201. [Online; accessed 2-September-2022]. 2018.
— *GPhoto — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=GPhoto&oldid=1097546275. [Online; accessed 2-September-2022]. 2022.
— *Gpsd — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Gpsd&oldid=1076207218. [Online; accessed 2-September-2022]. 2022.
— *Grafana — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Grafana&oldid=1104046322. [Online; accessed 30-August-2022]. 2022.
— *Gratis versus libre — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Gratis_versus_libre&oldid=1085552810. [Online; accessed 30-August-2022]. 2022.
— *Instrument Neutral Distributed Interface — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Instrument_Neutral_Distributed_Interface&oldid=1062506145. [Online; accessed 31-August-2022]. 2021.

— *International Telecommunication Union — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=International_Telecommunication_Union&oldid=1105915792. [Online; accessed 28-August-2022]. 2022.
— *KStars — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=KStars&oldid=1076667045. [Online; accessed 2-September-2022]. 2022.
— *Lagrange point — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Lagrange_point&oldid=1102894784. [Online; accessed 2-September-2022]. 2022.
— *Linux kernel — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Linux_kernel&oldid=1105840074. [Online; accessed 28-August-2022]. 2022.
— *Matrix (protocol) — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Matrix_(protocol)&oldid=1106244486. [Online; accessed 28-August-2022]. 2022.
— *NASA — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=NASA&oldid=1107567899. [Online; accessed 30-August-2022]. 2022.
— *NORAD — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=NORAD&oldid=1105457081. [Online; accessed 30-August-2022]. 2022.
— *Open source — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Open_source&oldid=1103126225. [Online; accessed 28-August-2022]. 2022.
— *Open-source hardware — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Open-source_hardware&oldid=1105615869. [Online; accessed 30-August-2022]. 2022.
— *Open-source software — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Open-source_software&oldid=1104440334. [Online; accessed 30-August-2022]. 2022.
— *OpenCV — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=OpenCV&oldid=1106133881. [Online; accessed 2-September-2022]. 2022.
— *Operating system — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Operating_system&oldid=1104687084. [Online; accessed 3-September-2022]. 2022.
— *Orbit — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Orbit&oldid=1106406646. [Online; accessed 30-August-2022]. 2022.
— *Perturbation (astronomy) — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Perturbation_(astronomy)&oldid=1105431363. [Online; accessed 30-August-2022]. 2022.
— *Photon — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Photon&oldid=1104709214. [Online; accessed 2-September-2022]. 2022.
— *Pipeline (computing) — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Pipeline_(computing)&oldid=1096885455. [Online; accessed 2-September-2022]. 2022.
— *Pixel binning — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Pixel_binning&oldid=1085651660. [Online; accessed 4-September-2022]. 2022.

Wikipedia contributors. *Python (programming language) — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Python_(programming_language)&oldid=1107007073. [Online; accessed 30-August-2022]. 2022.

— *Radio frequency — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Radio_frequency&oldid=1104615064. [Online; accessed 28-August-2022]. 2022.

— *Radio masts and towers — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Radio_masts_and_towers&oldid=1103964392. [Online; accessed 28-August-2022]. 2022.

— *Raspberry Pi — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Raspberry_Pi&oldid=1107949674. [Online; accessed 2-September-2022]. 2022.

— *Satellite — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Satellite&oldid=1106286374. [Online; accessed 30-August-2022]. 2022.

— *Satellite Catalog Number — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Satellite_Catalog_Number&oldid=1100318368. [Online; accessed 4-September-2022]. 2022.

— *Simplified perturbations models — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Simplified_perturbations_models&oldid=983925578. [Online; accessed 30-August-2022]. 2020.

— *Slewing — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Slewing&oldid=1061997651. [Online; accessed 2-September-2022]. 2021.

— *Software repository — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Software_repository&oldid=1081326130. [Online; accessed 2-September-2022]. 2022.

— *Software-defined radio — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Software-defined_radio&oldid=1107093398. [Online; accessed 28-August-2022]. 2022.

— *Star catalogue — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Star_catalogue&oldid=1107326057. [Online; accessed 2-September-2022]. 2022.

— *Star chart — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Star_chart&oldid=1106250278. [Online; accessed 2-September-2022]. 2022.

— *Star trail — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Star_trail&oldid=1075094531. [Online; accessed 2-September-2022]. 2022.

— *Teledyne FLIR — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Teledyne_FLIR&oldid=1102704879. [Online; accessed 3-September-2022]. 2022.

— *Telemetry — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Telemetry&oldid=1080706265. [Online; accessed 30-August-2022]. 2022.

— *Telescope — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Telescope&oldid=1106167988. [Online; accessed 30-August-2022]. 2022.

— *Toolchain — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Toolchain&oldid=1094527160. [Online; accessed 2-September-2022]. 2022.
— *Two-line element set — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Two-line_element_set&oldid=1104028347. [Online; accessed 28-August-2022]. 2022.
— *Ultra high frequency — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Ultra_high_frequency&oldid=1096417717. [Online; accessed 28-August-2022]. 2022.
— *Ultrasonic motor — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Ultrasonic_motor&oldid=1099892445. [Online; accessed 4-September-2022]. 2022.
— *Unix — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Unix&oldid=1105826457. [Online; accessed 2-September-2022]. 2022.
— *Upstream (software development) — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Upstream_(software_development)&oldid=1066370459. [Online; accessed 3-September-2022]. 2022.
— *Very high frequency — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Very_high_frequency&oldid=1105564543. [Online; accessed 28-August-2022]. 2022.

# Index


.cal, 90, 91
.cat, 90, 91
.csv, 90, 94–96
.dat, 90, 93, 94
.fits, 90, 91
.fits.png, 92
.png, 90
/dev/ttyACM, 52
/dev/ttyUSB, 56
/dev/video, 56

acquire, xiii, 28, 46, 47, 62, 63, 81, 90
airplane, 81–85, 93, 94
alignment, 43, 55
allsky, 30, 31
altazimuth mount, 22
altitude, 81
amateur radio, 16, 51
analyze, 46
antenna, 22, 35, 43, 44, 51, 52
application programming interface (API), 117
appropriate technology, 115
ARM64, 35
astap, 66
astometry.net, 70
astrograph, 30, 44, 107
AstroImageJ, 70
astrometry.net, 48, 66
astronomical catalogue, 117, 118

bashrc, 48
Basic Input/Output System (BIOS), 113
Bassa, Cees, 91
bcsak, 17
BeiDou Navigation Satellite System (BDS), 52
Berkeley Software Distribution (BSD), 104
binning, 90
Bosch, 35, 38, 40

C, xiv, 31, 74
camera, xii, 22, 28, 34–44, 51, 54, 62
Canon, 30, 35
Celestron, 42, 43, 51
central processing unit (CPU), 112
cgps, 53, 56
charge-coupled device (CCD), viii, 44, 55, 56, 59, 112, 114
China, 52
Comma Separated Value (CSV), 74, 90, 94
Commodity off the shelf (COTS), 43, 52, 112
community, 98
Complementary metal–oxide–semiconductor (CMOS), 38
config_new.ini, 70
configuration.ini, 47, 62, 70
COSPAR, 48, 94, 96
CubeSat, viii, 16–20, 117

daemon, 52, 114
dashboard, 17
data, 90
Debian, 30, 38, 46, 47, 52–54, 103
Debian Free Software Guidelines (DFSG), 30, 34, 48
decay, 17
detect, xiii, 28, 66, 70, 92
digital single-lens reflex (DSLR), 30, 34

Direct current (DC), 38
distributed control system (DCS), 105
Docker, 46, 60

Ekman, Daniel, 60
Ekos, 30, 42, 51, 54, 56, 66
embedded system, 22, 28, 34, 35, 46, 47, 52, 55, 56, 81, 107, 112, 113, 116
eMMC, 35
enclosure, 22, 25, 34, 35, 37–42
equatorial (EQ), 43, 55, 56, 113
ethernet, 34, 38, 40
Europe, 52

fan, 38, 40
Field of View (FoV), 30, 43, 81, 92
firmware, 44
FITS, 62
Flexible Image Transport System (FITS), 28, 30, 54, 66, 70, 74, 81, 90, 91
focus, 55
fork, 43, 48
formats, 90
forum, 98
Forward-looking infrared (FLIR), 118
four frame, 91
frame size, 90
free and open-source software (FOSS), 104, 106
Free open-source hardware (FOSH), 106
Free Software, 103, 104, 114, 115
Free/libre and open-source software (FLOSS), 104, 105

G-5500, 42
Galileo, 52
German EQ, 43
gigahertz (GHz), 107, 108
Giza, 46, 48, 74
Global Navigation Satellite System (GLONASS), 52
Global Navigation Satellite System (GNSS), 35, 43, 52, 53, 55, 81
Global Positioning System (GPS), 35, 52, 53, 114
GNU General Public License (GPL), 104
GNU's Not Unix! (GNU), 38, 112, 115
GoTo, 43, 54
gPhoto, 30, 31
gpsd, 52, 53
gpsmon, 53
Grafana, 17
ground station, viii, xii, 16, 17, 22, 23, 25, 28, 34, 35, 46, 51, 62, 107

hamlib, 42, 51
hardware, xiii, 28, 34, 35, 40
Hardware description language (HDL), 106
help, 98
hertz (Hz), 91
hough, 94, 95
hough3d-code, 46

identify, xiii, 28, 70, 74, 81, 92–94
IMX174, 34, 38
INDIGO, 51
indiserver, 56
Ingress Protection code 67 (IP67), 35
Instrument Neutral Distributed Interface (INDI), 30, 42, 51, 54–56
Intel, 35
International Space Station (ISS), 44, 112
International Telecommunication Union (ITU), 108, 109
Internet Protocol Address (IP), 52, 55
Internet Protocol version 6 (IPv6), 52
IOD, 94
iOptron, 42, 44

kilohertz (kHz), 107
Kowa, 35, 38, 40
KStars, viii, 30, 42, 43, 51, 54–59, 66

Lagrange point, 115
latitude, 81
lens, 34, 35
libre, 17, 34, 48, 66, 104, 106, 113, 115
Libre Space Foundation (LSF), xii, xiii, 17, 22, 31, 74, 98, 117
Linux, 30, 38, 104, 112
locate, 74
longitude, 81

lsusb, 56

mast, 22
Matrix, 98
megahertz (MHz), 108, 109
Mobile Industry Processor Interface (MIPI), 30
motion video, 30, 31
mount, 22, 25, 30, 34–37, 40, 42–44, 50, 51, 54, 56
mount plate, 40

National Aeronautics and Space Administration (NASA), 108
Network Time Protocol (NTP), 43, 53, 55, 81
New General Catalogue of Nebulae and Clusters of Stars (NGC), 113
Next Unit of Computing (NUC), 35
Nikon, 30
noise, 81
North American Aerospace Defense Catalog Number (NORAD ID), ix, 74, 92–94
North American Aerospace Defense Command (NORAD), 108
ntpd, 54
ntpdate, 54
numpy, 47

observation, 17, 19, 20
Odroid, 30, 35, 38, 40, 46, 56
Open Source, xiii, 104, 106, 111, 114
open-source appropriate technology (OSAT), 111
open-source hardware (OSH), xiii, 17, 104
open-source software (OSS), xiii, 17, 104
OpenCV, 30
Operating System (OS), 104, 112, 113, 115, 119
operation, 46, 62
optical ground station, xii, 22, 34
orbit, 17, 44, 48, 74, 92, 106, 112, 114, 117

Package Installer for Python (PIP), 47
Pan-tilt-zoom (PTZ), 42, 51
Personal Computer (PC), 107, 113, 115
perturbation, 108
pgplot, 48
photon, xii
PiCamera, 30, 34
pier, 40
PIP, 46
pipeline, 53
plate solver, xii, 28, 31, 66, 92
Platform as a Service (PaaS), 113
Portable Network Graphics (PNG), ix, 74, 90, 92–95
power cable, 40
Power over Ethernet (PoE), 35, 38
power supply, 34
predict, 94, 95
Printed Circuit Board (PCB), 106
process, xiii, 29, 46, 67, 70, 74, 76, 81, 90
proprietary, 30, 34, 48, 66
Pulse per second (PPS), 53
Python, xiv, 30, 31, 47, 70, 117

radio frequency (RF), xii, xiii, 17, 22, 51
RamSat, 16–20
Random-access memory (RAM), 48
Raspberry Pi, 30, 35, 40, 46
receiver, 22
repository, 46
rotator, 22, 42, 44, 51
Rowe-Ackermann Schmidt Astrograph (RASA), 30, 44
Russia, 52

sample rate, 90
satellite, xii–xiv, 16, 17, 28, 30, 31, 43, 44, 48, 54, 70, 74, 104, 108, 111, 112, 114, 115, 117, 118
satid, 74
SatNOGS, viii, xii, xiii, 16, 17, 22, 23, 30, 46, 51, 90, 107, 114, 116, 117
SatNOGS DB, viii, 16, 18
SatNOGS Optical, viii, xii, xiii, 22, 25, 28–31, 34, 35, 46, 62
satpredict, 46
sattools, 31, 48, 74, 112, 117
schedule, 17

SExtractor, 47
Simple Imaging Polynomial (SIP), 67, 116
Simplified Deep Space Perturbations (SDP), 17, 108
Simplified General Perturbations (SGP), 17, 107
Single board computer (SBC), 116
sky chart, viii, 51, 54, 56, 57
Sky-Watcher, 22, 35, 42, 51, 54, 56
skymap, viii, 48–50
slew, 43, 56, 81
software, xiii, 28, 46, 50, 54
Software development kit (SDK), 30
software repository, 46, 47
Software-defined radio (SDR), 22, 114
solar power, 35
Source Extractor, 46–48, 66
Space Defense Squadron (SDS), 106
Space Situational Awareness (SSA), 74
Spacecraft Identification and Localization (SIDLOC), 74
star catalogue, 66, 111, 116
star trail, xii, 42, 44, 54
Stellarium, 51
still camera, 30, 31
stphot, 31
stvid, ix, 31, 42, 43, 47, 54, 56, 62, 66, 67, 70, 74, 81, 90–96, 116, 117
support, xiii, 98
systemctl, 53
systemd, 52

Teledyne FLIR, 42
telemetry, viii, 17, 18, 114
telescope, 22, 30, 31, 35, 37, 38, 40, 42–44, 51, 54, 56
The Imaging Source, 30, 34, 38, 40, 56
threshold, 94, 96
TLE, 48, 70
toolchain, xii, 30, 48, 54, 55, 70
track, 22, 30, 34, 35, 40, 42–44, 50, 51, 54, 56
tripod, 22, 25, 34, 35, 40
two-line element set (TLE), xiii, 17, 28, 43, 44, 62, 81, 92, 93, 108

U-Blox, 52
Ultra High Frequency (UHF), 17, 22, 109
Ultrasonic motor (USM), 35
unidentified, 74, 77–80, 93, 94
United States Space Command (USSPACECOM), 106
Universal Serial Bus (USB), 34, 38, 52, 55
Unix, 112, 115
upload, xiii, 46, 90
UPSat, xiii, 17, 19
upstream, 48
USA, 52
USB video device class (UVC), 30

v4l2-ctl, 56
Very High Frequency (VHF), 22, 108
Video for Linux v2 (V4L2), viii, 30, 55, 58
virtualenv, 46, 47, 62
Volt (V), 38

Watney, 66
weather, 40
wifi, 34
World Coordinate System (WCS), 67, 116

Yaesu, 42

ZWO ASI, 30, 34

# *Colophon*

Created with 100% Free Software

Debian GNU/Linux

LaTeX