Astrelle Sky Simulator

Technical Manual and User Guide

Author: Srujan Kulkarni

Affiliation: RV College of Engineering, Bengaluru

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Modern astronomy relies on the acquisition of digital images of the sky using sensitive detectors, typically Charge-Coupled Devices (CCDs) or CMOS sensors. The standard data format for these images is the Flexible Image Transport System (FITS), a digital file format that stores not only the multi-dimensional pixel data but also a wealth of metadata in a human-readable header.

The recent and rapid deployment of satellite mega-constellations in Low Earth Orbit (LEO) has introduced a significant and pervasive source of contamination in these astronomical images. When a satellite passes through a telescope's field of view during an observation, its reflected sunlight creates a bright, linear trail. These trails can obscure or saturate the light from faint astronomical sources, corrupt photometric measurements, and trigger false positives in automated searches for transient events like supernovae or asteroids [6].

The development and validation of algorithms to detect and mitigate these trails require large, diverse, and well-characterized datasets. Astrelle is a high-fidelity, cross-platform sky simulation software designed to generate such datasets. It creates realistic synthetic FITS images of the sky by modeling the entire photon-to-pixel pipeline from first principles, including high-precision **SGP4 orbit propagation** for satellites. Astrelle provides an invaluable "ground-truth" dataset for training and testing detection algorithms. This manual details the software's functionality, the physical models it employs, and its implementation.

Chapter 2

Installation

Astrelle is distributed as a self-contained application for Linux, Windows, and macOS. No manual installation of dependencies (like Python or system libraries) is required.

2.1 Downloading the Application

Download the correct package for your operating system from the latest release on the official GitHub repository:

https://github.com/IonizedSrujan/Astrelle/releases

2.2 Linux (AppImage)

The Linux version is provided as an AppImage, which runs on most modern distributions (e.g., Ubuntu, Fedora, Debian).

1. Download the ‘astrelle_*.AppImage‘ file.
2. Make the file executable. You can do this through your file manager’s properties menu or in the terminal:

```
chmod +x astrelle_*.AppImage

```

3. Double-click the file to run it, or execute it from the terminal:

```
./astrelle_*.AppImage

```

2.3 Windows

The Windows version is a standard installer executable.

1. Download the ‘astrelle_*.exe‘ file.

2. Double-click the installer and follow the on-screen instructions to complete the installation.

3. After installation, launch the "Astrelle Sky Simulator" from your Start Menu or the desktop shortcut.

2.4 macOS

The macOS version is provided as a standard application bundle inside a zip archive.

1. Download the ‘astrelle_*_macOS.zip‘ file.

2. Extract the archive to get the ‘Astrelle.app‘ bundle.

3. Move ‘Astrelle.app‘ to your Applications folder.

4. The first time you open the application, you may need to right-click it, select "Open", and confirm that you trust the application in the security dialog that appears.

Chapter 3

User Guide

This chapter provides a detailed walkthrough of the Astrelle Graphical User Interface (GUI). We will cover each component of the main window and explain how to configure and run a simulation.

3.1 Main Window Overview

When you launch Astrelle, you are presented with the main window, which is organized into three key areas for an efficient workflow.

- **Control Panel (Left):** This is where you configure all simulation parameters. Settings are grouped into collapsible sections.
- **Image Viewer (Top Right):** After a simulation is complete, the final FITS image is displayed with enhanced contrast for clear viewing.
- **Log Window (Bottom Right):** This window provides real-time feedback, status updates, query results, and error messages during the simulation process.

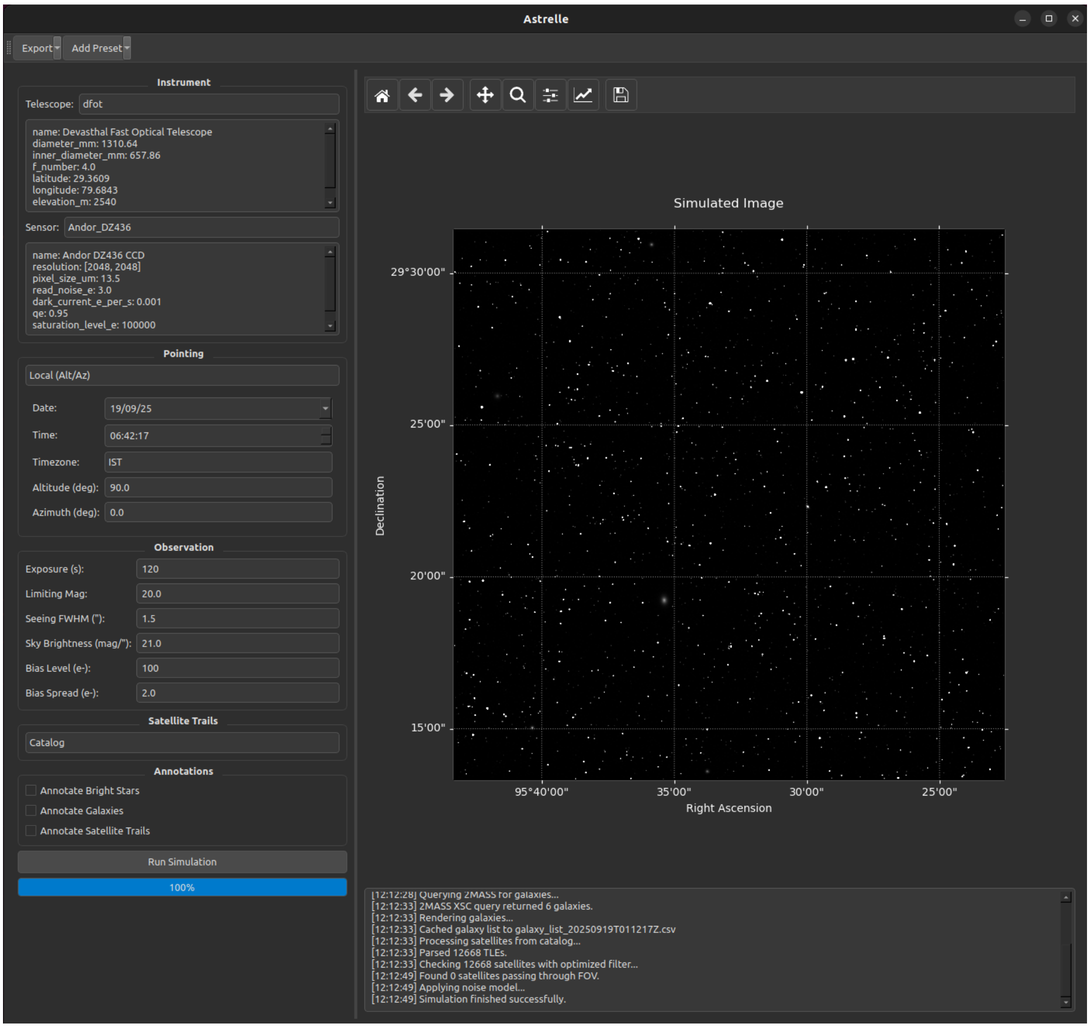

Figure 3.1: The main window of the Astrelle Sky Simulator, showing the Control Panel, Image Viewer, and Log Window.

3.2 The Control Panel: Configuring a Simulation

The control panel on the left contains all the settings required to define your simulation. These are logically grouped into collapsible sections.

3.2.1 Instrument Configuration

This section allows you to select the telescope and sensor combination for the simulation.

- **Preset:** Choose from a list of pre-defined instrument setups. Astrelle uses a JSON file (`presets.json`) to manage these. The default preset is for the Devasthal Fast Optical Telescope (DFOT).

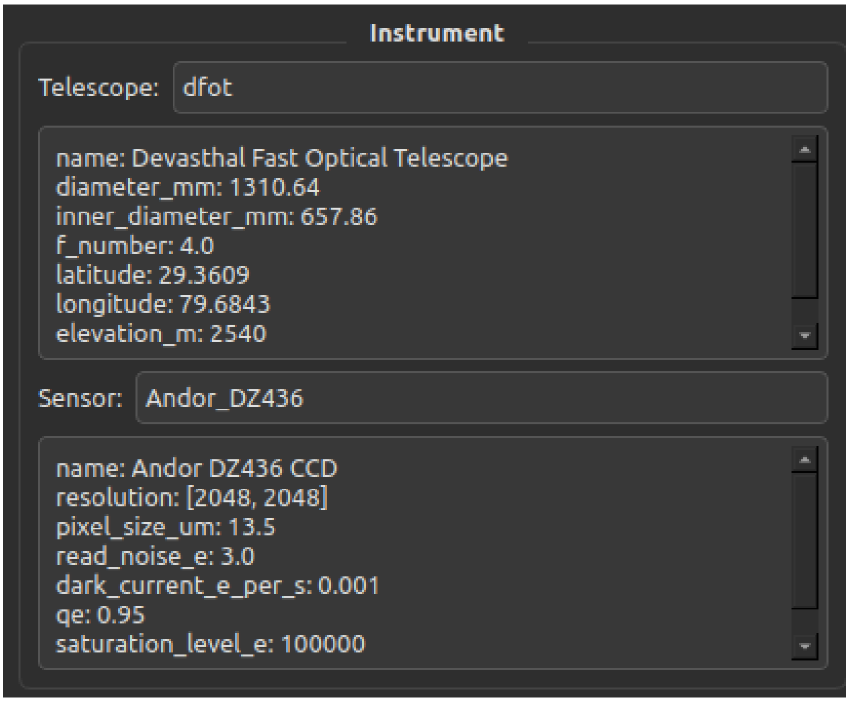

Figure 3.2: The Instrument Configuration section. Select a preset from the dropdown menu.

3.2.2 Pointing and Observation

Here, you define where the telescope is pointing and the core parameters of the observation.

- **Pointing Coordinates:** You can specify the center of the field of view in two ways: RA/Dec or Alt/Az.
- **Observation Parameters:** Configure exposure time, atmospheric seeing, and sky brightness.

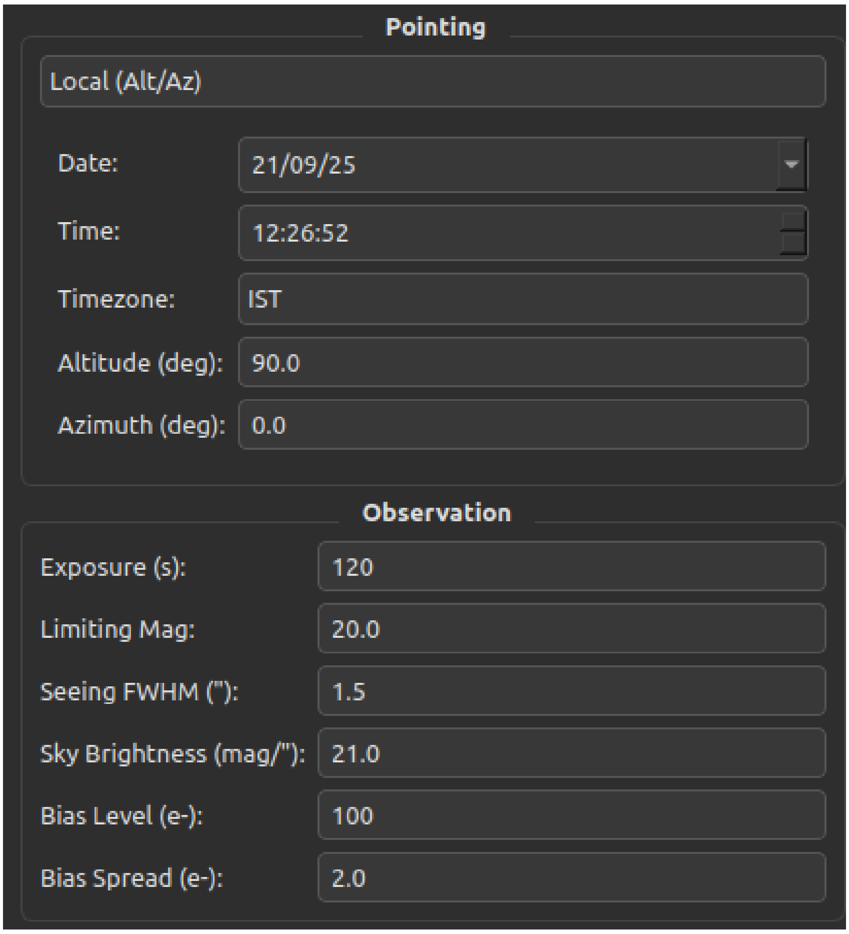

Figure 3.3: The Pointing and Observation section, where the telescope's target and exposure settings are defined.

3.2.3 Satellite Simulation

This section controls how satellite trails are generated.

- **Simulation Mode:**
  - **Catalog Mode:** This is the default mode. Astrelle downloads the latest Two-Line Element (TLE) data and simulates trails for real satellites. A dropdown menu allows you to select from various CelesTrak catalogs, including **Starlink, OneWeb, GPS, GLONASS, Galileo, weather satellites, and the ISS**. You can also export the downloaded TLEs for the selected group to a '.txt' file.
  - **Synthetic Mode:** This mode allows you to create artificial satellite trails. You can choose to generate trails with random parameters or define one manually by its start pixel, velocity, and apparent magnitude. This is useful for testing detection algorithms with precisely known trail characteristics.

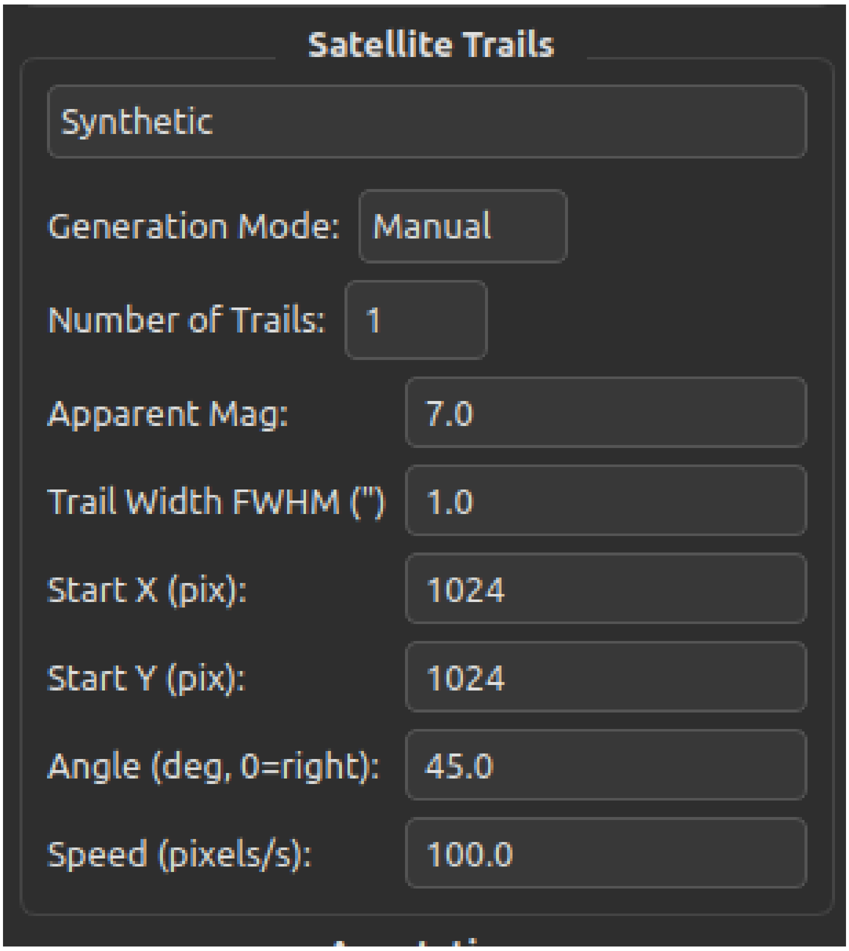

Figure 3.4: The Satellite Simulation section, allowing a choice between real-world (Catalog) and user-defined (Synthetic) trails.

3.2.4 Running the Simulation and Viewing Output

Once all parameters are set, you can start the simulation.

- **Run Simulation Button:** Click this button to begin. The button will change to a **Stop Simulation** button, allowing you to gracefully interrupt a long simulation at any time.
- **Monitoring Progress:** Watch the **Log Window** for real-time updates.
- **Viewing and Exporting:** Upon completion, the image appears in the viewer. You can export the FITS file, a PNG preview, and CSV ground-truth tables.

Chapter 4

Tutorial: Simulation Examples

This chapter provides step-by-step examples for common use cases.

4.1 Example 1: Simulating a Star Field using Local Coordinates

Let's perform a simple 60-second simulation pointing at the eastern sky using local Altitude/Azimuth coordinates.

1. **Instrument:** Select the default `DFOT` preset.
2. **Pointing:** Choose the **Alt/Az** coordinate system. Set Altitude to `30` and Azimuth to `90`.
3. **Observation:** Set Exposure Time to `60` s, Seeing to `1.8` arcsec, and Sky Brightness to `20.0` mag/arcsec^2.
4. **Satellites:** Keep the default **Catalog Mode** selected. For this example, choose the "Starlink" satellite group from the dropdown menu.
5. **Run:** Click **Run Simulation**.

Expected Result: The simulator will generate an image of the stars in that patch of sky and will overlay a trail if any Starlink satellites are predicted to pass through the field during the 60-second exposure.

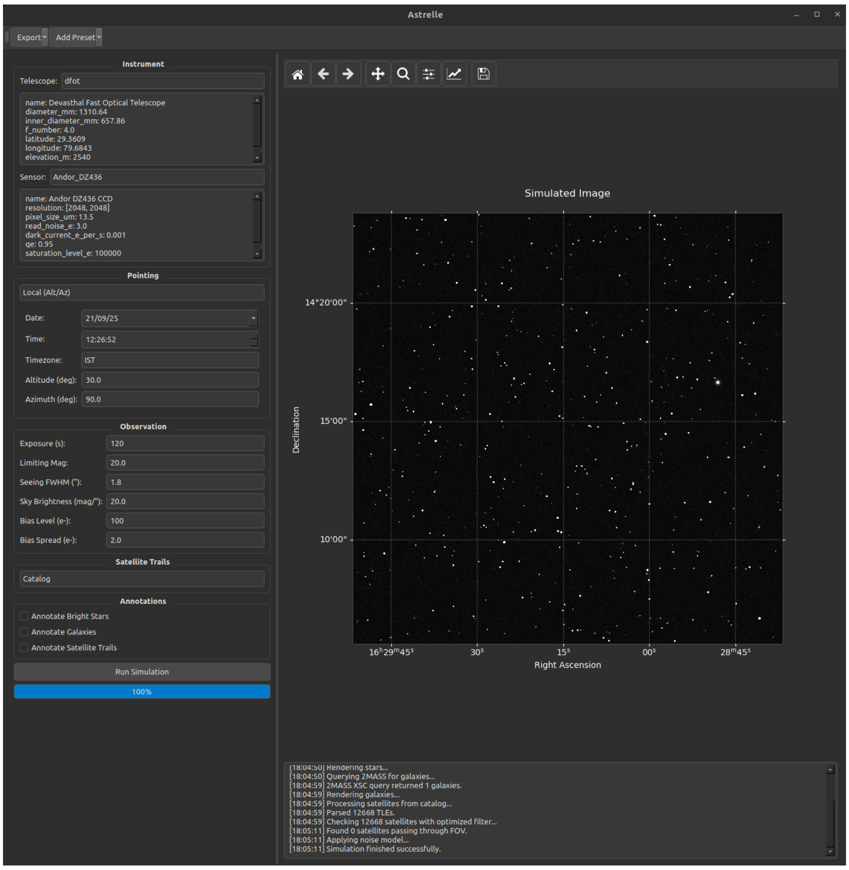

Figure 4.1: Expected output for Example 1: A star field corresponding to the specified local coordinates.

4.2 Example 2: Creating a Synthetic Satellite Trail

This example simulates a field and adds a user-defined, artificial satellite trail. This is ideal for testing trail detection algorithms.

1. **Instrument:** Select the default `DFOT` preset with any sensor you prefer.
2. **Pointing:** Select the **RA/Dec** coordinate system. Set RA to `270` and Dec to `29`.
3. **Observation:** Set Exposure Time to `300` s, Seeing to `1.2` arcsec, and Sky Brightness to `22.0` mag/arcsec^2 (for dark sky conditions).
4. **Satellites:** Change the Simulation Mode to **Synthetic Mode**.
5. **Configure Synthetic Trail:** Choose the "Random" mode for trail generation.
6. **Run:** Click **Run Simulation**.

Expected Result: The image will show a star field with a synthetic trail whose position, angle, and brightness have been randomly generated within realistic bounds.

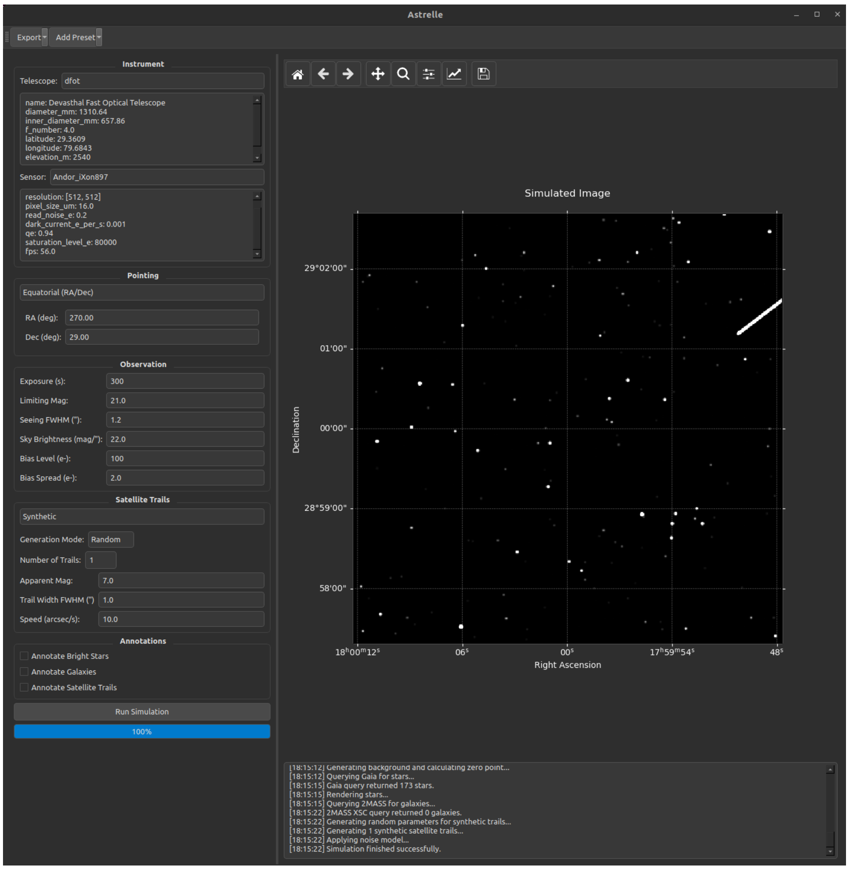

Figure 4.2: Expected output for Example 2: A deep field with a randomly generated synthetic trail.

Chapter 5

Technical Reference: The Physical Model

The Astrelle simulator is built as a modular pipeline, where each stage represents a distinct physical process in the journey of light from a celestial object to a final data number in a FITS file.

Table 5.1: Key Parameters for the Devasthal Fast Optical Telescope (DFOT) Setup

Parameter	**Value**
Telescope Aperture	1.3 m
Focal Ratio	f/4
Observatory Latitude	29.3609° N
Observatory Longitude	79.6843° E
Observatory Elevation	2540 m
Default Sensor	Andor DZ436
Sensor Resolution	2048 x 2048 pixels
Pixel Size	13.5 μm

5.1 Radiometric Calibration: The Instrumental Zero-Point

The foundation of a physically accurate simulation is a robust radiometric calibration. We must establish a precise mathematical relationship between the standard astronomical magnitude system and the physical number of photo-electrons generated in the detector. This is achieved by calculating the instrumental **zero-point** (Z_p), defined as the count rate (in electrons per second, e^-/s) that would be measured from a star of magnitude 0. The relationship is given by:

$$N_{e^-} = t_{exp} \cdot Z_p \cdot 10^{-0.4m} \tag{5.1}$$

This value is derived by numerically integrating the spectral flux of a standard star (Vega) through the modeled atmospheric and instrumental transmission functions [3,10], ensuring a physically grounded conversion from magnitudes to electron counts.

5.2 Sky Background and Stellar Population

The stellar population is sourced from the **Gaia DR3** catalog [5]. An ideal star is a perfect point source. The turbulence in the Earth's atmosphere blurs this point into a fuzzy spot via a process called **convolution**. The resulting shape is the **Point Spread Function (PSF)**, which is modeled as a 2D Gaussian profile based on the user-defined "seeing" value [16].

5.3 Galaxy Population

Galaxies are extended sources sourced from the **2MASS Extended Source Catalog (XSC)** [13]. Their ideal light distribution is modeled using the Sersic profile [11]. To simulate the final on-sky appearance, this ideal Sersic profile is **convolved** with the same atmospheric Gaussian PSF used for stars, realistically blurring the galaxy's features.

5.4 Satellite Simulation: Dynamics, Photometry, and Rendering

The simulation of satellites is a three-stage process involving precise orbital prediction, a physical brightness model, and a continuous rendering algorithm.

5.4.1 Orbital Propagation with SGP4

The simulation's predictive power comes from the Standard General Perturbations 4 (**SGP4**) model, the de facto standard for propagating the orbits of Earth-orbiting objects from their Two-Line Element (TLE) sets [18]. Astrelle implements this via the Skyfield library, ensuring high-fidelity positional accuracy. TLE data is sourced in near real-time from CelesTrak for numerous constellations, including Starlink, OneWeb, GPS, and others. For each time step in the simulation, SGP4 calculates the satellite's precise position and velocity vectors, which are then projected onto the celestial sphere and the detector plane.

5.4.2 Satellite Photometry Model

Instead of using a fixed brightness, Astrelle calculates a satellite's apparent visual magnitude (m_v) based on a physical model of reflected sunlight [9]. The magnitude is determined by:

$$m_v = m_\odot - 2.5 \log_{10} \left(\frac{A \cdot p \cdot \Phi(\alpha)}{D^2} \right) + C \tag{5.2}$$

where:

- $m_\odot$ is the apparent magnitude of the Sun (-26.74).
- A is the satellite's effective cross-sectional area (an assumed parameter).
- p is the satellite's geometric albedo (reflectivity).
- D is the distance from the observer to the satellite in astronomical units.
- $\Phi(\alpha)$ is the phase function, which describes how the satellite's brightness changes with the phase angle α (the Sun-satellite-observer angle). For a diffusely reflecting sphere, this is given by $\Phi(\alpha) = \frac{2}{3\pi}[(\pi - \alpha)\cos(\alpha) + \sin(\alpha)]$.
- C is a calibration constant.

This model ensures that a satellite's brightness realistically changes as its geometry relative to the Sun and observer changes during a pass.

5.4.3 Continuous Trail Rendering

A satellite trail is not a series of dots but a continuous streak. The rendering algorithm simulates this by treating the satellite's path during a short time interval Δt as a straight line segment on the detector. The total flux from the satellite during this interval is calculated and then distributed along this line. The width of the trail is generated by convolving this line segment with the atmospheric PSF. For any given pixel near the trail, its final value is an integral of the flux from the portion of the trail that passed over it, blurred by the seeing profile. This method produces a smooth, continuous trail with a realistic brightness profile.

5.5 Detector Noise Model

The final stage is to convert the ideal, noiseless floating-point image into a realistic integer-valued FITS image by simulating the primary stochastic noise sources inherent in detectors [7]:

$$\sigma_{\text{pixel}} = \sqrt{N_{\text{signal}} + N_{\text{dark}} + \sigma_{\text{read}}^2} \tag{5.3}$$

This includes simulating Poissonian **shot noise** on the signal and dark current, and adding Gaussian **read noise** from the electronics.

Chapter 6

Results and Discussion

6.1 The Astrelle Simulator: Software and Functionality

The primary result of this project is the Astrelle software itself, a fully functional simulation tool capable of producing high-fidelity synthetic astronomical images. It is controlled via a comprehensive Graphical User Interface (GUI) as shown in Figure 3.1, which provides an intuitive workflow for setting up and running complex simulations.

6.2 Qualitative Analysis of Simulated Images

Visual inspection of the simulated FITS files reveals a high degree of realism. Key features that mimic real observations are immediately apparent, including realistic stellar PSFs, a noisy background texture, and a wide dynamic range.

Figure 6.2 demonstrates the core functionality of the simulator. A bright satellite trail is superimposed on the star field. The simulation accurately captures key features of a real satellite trail:

- **Linearity and Smoothness:** The trail is a sharp, linear, and continuous feature, consistent with the SGP4-predicted motion and the continuous rendering algorithm.
- **Width and Profile:** Its width is determined by the convolution of the satellite's path with the atmospheric PSF, giving it a realistic, soft-edged profile.
- **Flux Deposition:** The surface brightness of the trail correctly corresponds to the satellite's physically modeled apparent magnitude.

The rendering of extended sources like galaxies using elliptical Sersic profiles, as seen in Figure 6.3, adds another layer of realism. The combination of these elements results in a synthetic image that is challenging to distinguish from a real astronomical observation at first glance.

Figure 6.1: Example simulation of a dense star field in the galactic plane. Note the variation in stellar brightness and the noisy texture of the background.

Figure 6.2: A simulation showing a satellite trail passing through a moderately dense star field. The trail is rendered using the continuous algorithm for a smooth, realistic appearance.

Figure 6.3: A simulated deep field containing several galaxies rendered as Sersic profiles, convolved with the atmospheric PSF.

6.3 Quantitative Validation

Beyond visual inspection, the simulator's output must be quantitatively correct. This validation is twofold: **photometric** (are the brightnesses correct?) and **astrometric** (are the positions correct?).

6.3.1 Photometric Validation

The scientific validity of the simulation rests on the correctness of its physical models. The radiometric calibration pipeline ensures that the number of electrons generated for any object of a given magnitude is physically correct for the specified instrument and conditions. This ground-truth nature is what makes the generated data invaluable for algorithm testing.

6.3.2 Astrometric Validation

A critical requirement for any scientific image simulator is astrometric accuracy. To confirm this, a simulated FITS file was submitted to the "blind" astrometric solver,

Astrometry.net. The service analyzes the pattern of stars in an image and, without any prior information, determines the image's true pointing, orientation, and pixel scale.

The simulated image (input parameters in Figure 6.4) was successfully "solved." The solution from Astrometry.net (Figure 6.5) perfectly matched the input parameters. This test confirms that the simulator's gnomonic projection and WCS header generation are implemented correctly.

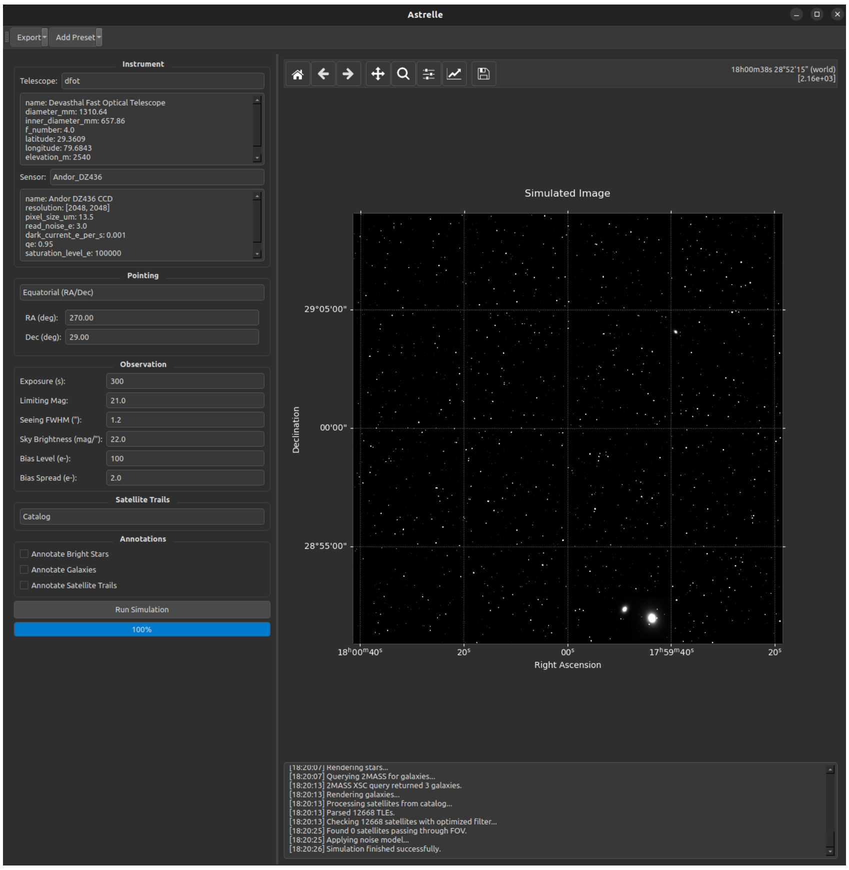

Figure 6.4: Input parameters for the astrometric validation test.

Calibration

Center (RA, Dec):	(270.000, 29.000)
Center (RA, hms):	$17^h\ 59^m\ 59.980^s$
Center (Dec, dms):	+29° 00' 00.275"
Size:	18.1 x 18.1 arcmin
Radius:	0.214 deg
Pixel scale:	0.531 arcsec/pixel
"Orientation", may be incorrect, use at your own risk:	Up is -0.0 degrees E of N
WCS file:	wcs.fits
New FITS image:	new-image.fits
Reference stars nearby (RA,Dec table):	rdls.fits
Stars detected in your images (x,y table):	axy.fits
Stars detected in your images, converted to RA,Dec (FITS table):	image-radec.fits
Correspondences between image and reference stars (table):	corr.fits
Legacy Surveys sky browser:	browse the sky
KMZ (Google Sky):	image.kmz
World Wide Telescope:	view in WorldWideTelescope

Figure 6.5: The corresponding Calibration from Astrometry.net. The solution perfectly matches the input parameters, validating the simulator's WCS.

Chapter 7

Limitations and Future Scope

7.1 Limitations of the Current Model

While Astrelle produces high-fidelity images, it is important to acknowledge the physical and instrumental effects that are not simulated, such as complex galaxy morphologies, intricate PSF shapes (e.g., diffraction spikes), rapid satellite brightness variability ("glints"), and detector artifacts like cosmic rays.

7.2 Future Scope

The future scope involves two main thrusts: refining the simulator and developing an integrated trail detection system.

1. **Simulator Refinement:** The next iteration will focus on addressing current limitations, such as incorporating more complex PSF models and simulating optical distortions that require higher-order polynomial terms in the WCS solution.

2. **Integrated Trail Detection Module:** The primary purpose of this data is to train a robust trail detection algorithm. We will explore methods based on classical transform models, such as the Radon and Hough transforms, which are computationally efficient and highly effective for linear feature detection [15, 17].

Bibliography

[1] ARIES, *3.6m Devasthal Optical Telescope*, https://www.aries.res.in/facilities/astronomical-telescopes/36m-dot. Accessed 2025.

[2] Astropy Collaboration, et al. 2018, AJ, 156, 123, *The Astropy Project: Building an Open-science project and status of the v2.0 core package.*

[3] Bessell, M., & Murphy, S. 2012, PASP, 124, 140, *Spectrophotometric Libraries, Revised Photonic Passbands, and Zero Points for UBVRI, Hipparcos, and Tycho Photometry.*

[4] Kelso, T.S., CelesTrak. https://celestrak.com. Accessed 2025.

[5] Gaia Collaboration, et al. 2023, A&A, 674, A1, *Gaia Data Release 3: Summary of the content and survey properties.*

[6] Hasan, I., Tyson, J. A., Saunders, C., & Xin, B. 2023, AJ, 165, 237, *Mitigating Satellite Trails: A Study of Residual Light After Masking.*

[7] Howell, S. B. 2006, *Handbook of CCD Astronomy* (2nd ed.; Cambridge: Cambridge Univ. Press).

[8] Karttunen, H., Kröger, P., Oja, H., et al. (eds.) 2007, *Fundamental Astronomy* (5th ed.; Berlin: Springer).

[9] Krag, W. E. 1974, MIT Lincoln Laboratory Technical Note, 1974-27, *Visible Magnitude of Typical Satellites in Synchronous Orbits.*

[10] Mohan, V., et al. (1999), BASI, 27, 335. *Atmospheric extinction measurements at Devasthal, Naini Tal.*

[11] Peng, C. Y., Ho, L. C., Impey, C. D., & Rix, H.-W. 2002, AJ, 124, 266, *Detailed Structural Decomposition of Galaxy Images.*

[12] Rhodes, B. (2019). Skyfield: High precision research-grade positions for planets and earth satellites. Astrophysics Source Code Library, ascl:1907.024.

[13] Skrutskie, M. F., et al. 2006, AJ, 131, 1163–1183, *The Two Micron All Sky Survey (2MASS).*

[14] Smith, M. J., Geach, J. E., Jackson, R. A., et al. 2023, MNRAS, 521, 1076, *Realistic galaxy image simulation via score-based generative models.*

[15] Stark, D. V., Grogin, N., Ryon, J., & Lucas, R. 2015, ACS Instrument Science Report 2015-05, *Improved Identification of Satellite Trails in ACS/WFC Imaging Using a Modified Radon Transform.*

[16] Stetson, P. B. 1992, in Astronomical Data Analysis Software and Systems I, eds. D. M. Worrall, C. Biemesderfer, & J. Barnes, ASP Conf. Ser., 25, 297, *DAOPHOT II.*

[17] Stoppa, F., Groot, P. J., Stuik, R., et al. 2023, A&A, 676, A11, *Automated detection of satellite trails in ground-based observations using U-Net and Hough transform.*

[18] Vallado, D. A., Crawford, P., Hujsak, R., & Kelso, T. S. 2006, AIAA/AAS Astrodynamics Specialist Conference, No. AIAA 2006-6753, *Revisiting Spacetrack Report #3.*