

---

# Astrometric Plate Solving of Images From The FUT

Samuel Grund Sørensen, Aarhus University, Denmark  
(Temporarily studying at University of Calgary, Alberta Canada)

---

December 15, 2023

## Abstract

Throughout its history, humanity has used constellations to navigate and identify seasonal changes. While modern clocks and navigation systems have largely removed the need for knowledge of the night sky, the systematic identification of star fields is still widely used in astronomy and satellite missions to identify and navigate the night sky. In this paper, an open-source algorithm for obtaining the transformation between image coordinates and physical, astronomical coordinates is discussed. It utilizes the same basic principles of matching star patterns, but considers arbitrary constellations (so-called asterisms) and applies Bayesian inference to evaluate the quality of the proposed transformations. To this end, the GAIA DR2 catalogue is used as a baseline index and a showcase of the solution of imagery of the Helix Nebula is provided. The presented algorithm is specialized to the specifications of a certain telescope, but is readily adaptable to other specifications with further work.

# 1 Introduction

The process of astrometric plate solving generally refers to the process of obtaining a *world coordinate system* (WCS) solution for certain astronomical image data. The WCS solution defines the transformation between instrumental image coordinates ( $x, y$ ) and absolute physical, celestial coordinates ( $a, b$ ). Typically, these physical coordinates will be the equatorial coordinates (RA, DEC), denoting a set of coordinates containing the *right ascension* (RA) and *declination* (DEC) of a certain object, but with appropriate complimentary information about the observation<sup>1</sup>, these coordinates can be transformed to whichever frame is best suited for subsequent analysis.

Plate solving sees immediate use in astronomical image analysis but is also relevant in the field of satellite *Attitude Determination and Control Systems* (ADCS). Typically satellites are able to orient themselves according to some internal coordinate system using instruments like magnetorquers or reaction wheels. To relate this satellite coordinate system to real-world directions, an absolute reference point is required. This can be provided by applying the technique to images of stars taken from the satellite.

Finally it should be noted that while this paper does not introduce radically new methods for approaching the problem of plate solving, it aims to highlight some of the key aspects of the method so as to demystify the process which is typically abstracted behind interfaces.

The code for this project can be found in the public Github repository at <https://github.com/samgrund/605-astrometric-solver>. The main notebook demonstrating the algorithms usage is `demo.ipynb`.

# 2 Methods

## 2.1 Algorithmic Overview

This section aims to provide a higher-level overview of the process of plate solving, specifically in the case of the presented implementation. Subsequent sections [2.2-2.6] describe the process components in higher detail. It should be noted that certain parameters of this process have been selected to suit images taken with the FUT<sup>2</sup>, which is a remote telescope associated with Aarhus University. With further work, the algorithm could relatively simply be adapted to accept a further range of image specifications.

In general, the process can be described by a black box that takes an image and its metadata as input, and returns a valid WCS. A certain very prevalent and standardized structure that contain both the image data and its metadata is a *Flexible Image Transport System* (FITS) file(Wells, Greisen, and Harten, 1981). This is the format this implementation is built to work with.

The first step of the process is to detect stars in the input image. From here, some sort of photometry is performed on the detected stars to extract an amount of relevant stars, evaluated by their brightness - brightest first. Having selected a subset of stars, polygons of connected stars (henceforth referred to as *asterisms*) are built according to

<sup>1</sup>E.g. time and location at which the observation is acquired.

<sup>2</sup>This is an abbreviation of the danish words "Fjernstyrede UndervisningsTeleskop", i.e. the remote educational telescope. Because of the foreign nature of the full name, it will only be referred to as FUT.

some geometric criteria. These asterisms are matched to an index catalogue of asterisms built from the same criteria, but with stars of known positions. By matching image and index asterisms via a so-called *hash*, the image coordinates of stars in the input image can be related to the physical positions of stars in the index. At this point, the WCS of the input image is well defined.

## 2.2 Star Detection

The process of extracting stars from an image can be described by grouping "outliers" in a background distribution of very low values. First, the image is roughly calibrated by subtracting the image median<sup>3</sup>. Then peaks are found by the criteria that their value must be greater than  $3\sigma$ , where  $\sigma$  is the standard deviation of the distribution of pixel values in the image. These peaks are then grouped by applying a 2D convolution with the  $3 \times 3$  "neighbor counter" kernel in eq. 1, and grouping connected peaks into star candidates.

$$k = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad (1)$$

Thus far there is no guarantee of the distribution of these peaks within a star candidate group. To evaluate their star-likeness, each group is fitted to a 2D gaussian profile. For some groups, this process will fail, which in practice acts as a filter to sort out groups that are too non-gaussian. From this process, a dataset describing the gaussian distribution parameters of each star is obtained. As a final filter to remove non-stellar detections, stars with  $\sigma$  parameters deviating more than  $3\sigma_\sigma$  from the median are rejected, where  $\sigma_\sigma$  refers to the standard deviation of the distribution of fitted  $\sigma$  parameters. Put in words, this process rejects stars that are much larger than the median star profile.

To select an appropriate subset of brightest stars to work on, aperture photometry is performed on the detected stars. In this process, an integration over pixel values within a certain radius of each star is performed. The radius chosen for this implementation corresponds to  $2m(\sigma)$ , where  $m(\sigma)$  refers to the median of stellar  $\sigma$  parameters.

The number of stars selected depends on the specified field density  $\rho_f$ , denoting the number of stars per square degree to consider. This parameter should in principle be updated dynamically according to the field size, but for the sake of this demonstration it has been set to  $\rho_f = 100/\text{deg}^2$ . For the FUT with a field size of  $30 \times 30$  arc minutes, this corresponds to  $\rho_f A_f = (30 \text{ arcmin} \times 30 \text{ arcmin}) \cdot 100/\text{deg}^2 = 25$  stars, where  $A_f$  is the field area.

## 2.3 Asterism and Hash Generation

To generate asterisms from the detected stars, each star is considered as an "origin star". A 4-star asterism is then constructed containing the origin star and 3 stars within a radius of 0.1 to 0.35 degrees of the origin star. This ensures that the asterisms have a certain angular extent that is not too small or large for the field size of the FUT.

At this point, asterisms are only defined by the image coordinates of their constituent stars, which is not rotationally invariant. To transform these asterisms into a new space, which is rotationally invariant, a new coordinate system is constructed by two basis

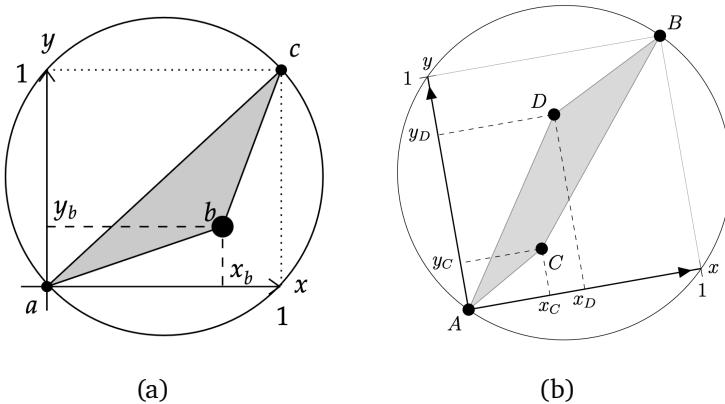
---

<sup>3</sup>The image median is much less sensitive to hot pixels and to skewness from the large contrast between the dark sky background and stars. Therefore it is a good candidate to evaluate the background level.

vectors  $x$  and  $y$  which are obtained by rotating the vector connecting the origin- and the furthest star 45° (anti)clockwise. By projecting each star coordinate onto these new basis vectors, a set of coordinates, which would have been the same if the input image had been rotated, is obtained. Because the origin star will always have projected coordinates  $(0, 0)$  this reduces the dimensionality of each asterism. Finally, each asterism is scaled by the scale factor of the image such that the new coordinates correspond to physical angular extents from the origin star.

The set of coordinates  $h = \{(x_c, y_c), (x_d, y_d), (x_b, y_b)\}$  constitutes the asterism *hash* which is the lowest dimensional rotationally invariant representation of the asterism.

This method only considers the relative positions of the stars. The originally proposed implementation suggested using not only the relative positions of stars, but also their relative brightnesses. This idea was rejected for two reasons: Firstly, the method proposed in section 2.2 favors using bright stars. The brightest stars in images are typically saturated so as to be able to resolve low-brightness objects better. Because the brightness of saturated objects cannot be reliably obtained, this is certainly not an appropriate hash parameter. Secondly, the brightness of a star is defined in a certain *band*. This band corresponds to a sensitivity function across a range of wavelengths, and relative brightnesses across different bands are not invariant. For these reasons, the strictly geometric approach was chosen instead.



**Figure 1:** (a): The proposed flux geometric hash where only 3 stars are used, but their relative brightness is considered. (b): The applied geometric hash using 4 stars (adapted from figure 1 of Lang et al., 2010).

## 2.4 Index Building

To obtain the WCS, hashes from image stars are matched to ones obtained from stars with known positions.

For this implementation, the GAIA DR2 catalogue (Brown et al., 2018) was selected. This database is easily accessible and contains the required information for this analysis. Furthermore, GAIA is a wide-field mission and so has observed stars across the entire sky. Finally, GAIA does not contain as dim sources as other more specialized missions, but for images from the FUT (and most other telescopes), this does not pose a problem.

Typically the *approximate* position of the field is known, and so the GAIA DR2 is queried for stars in a radius of  $2d$  of the approximate position, where  $d$  is the diagonal extent of the field. A number of stars, according to the field density  $\rho_f$ , is then selected for further processing.

To build an index of known asterisms, the procedure described in section 2.3 is followed, but instead of using stars from the image, the queried stars are used instead. Apart from this, the process is identical, and hence is not covered in further detail here.

## 2.5 Hash Matching

With a set of image asterism hashes and an appropriate index, asterisms can be matched. To do this, the  $\Delta^2$  parameter is introduced defined by

$$\Delta^2 = (x_{b,I} - x_{b,G})^2 + (x_{c,I} - x_{c,G})^2 + (x_{d,I} - x_{d,G})^2 + (y_{b,I} - y_{b,G})^2 + (y_{c,I} - y_{c,G})^2 + (y_{d,I} - y_{d,G})^2, \quad (2)$$

where  $x_{i,j}$  or  $y_{i,j}$  refers to the  $i$ th star in a hash and  $j$  refers to the hash source,  $I$  denoting the image and  $G$  denoting hashes built from GAIA. Geometrically similar hashes will have a low value of  $\Delta^2$  and therefore potential matches are ranked by this metric.

## 2.6 Solution Evaluation

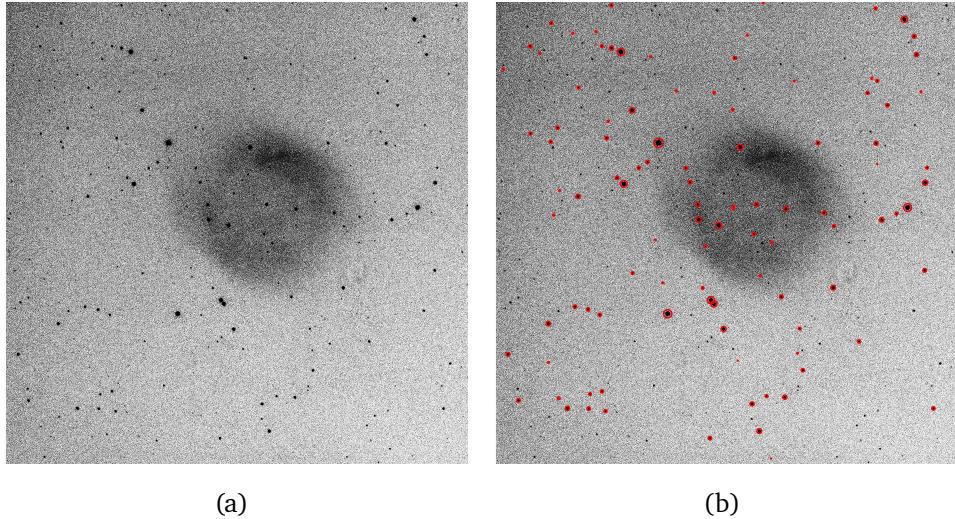
While the  $\Delta^2$  parameter is a good metric for the geometric similarity of two asterisms, it is not sufficient to guarantee an accurate WCS. Many phenomena give rise to a shift in the apparent position of a star, and therefore the correct match is not guaranteed to be the one with the lowest  $\Delta^2$ . Furthermore, because of the large number of stars, built asterisms may be geometrically similar simply by chance. Therefore, a more rigorous approach must be taken to ensure that the match is correct.

To do this, it is noted that each proposed solution holds predictive power. Given a proposed match (a hypothesis), an associated prediction of stellar positions exists. By this consideration, conventional hypothesis testing methods can be used to evaluate a match. In this implementation, the Bayes factor  $K = Pr(D|M_1)/Pr(D|M_2)$  will be used to evaluate how well the predictive power of a hypothesis performs compared to a null hypothesis in which the probability of finding a star *anywhere* is the same, representing a uniform 2D probability distribution.

Finally, if a match passes the criteria  $K > 10^6$ , it is marked as being statistically significant. From here, the WCS information can be calculated using the correspondence between image coordinates and physical coordinates given by the asterism match. In this implementation, the specifics of this step are abstracted into the `find_wcs_from_points` function from the Astropy library (Robitaille et al., 2013).

## 3 Results and Tests

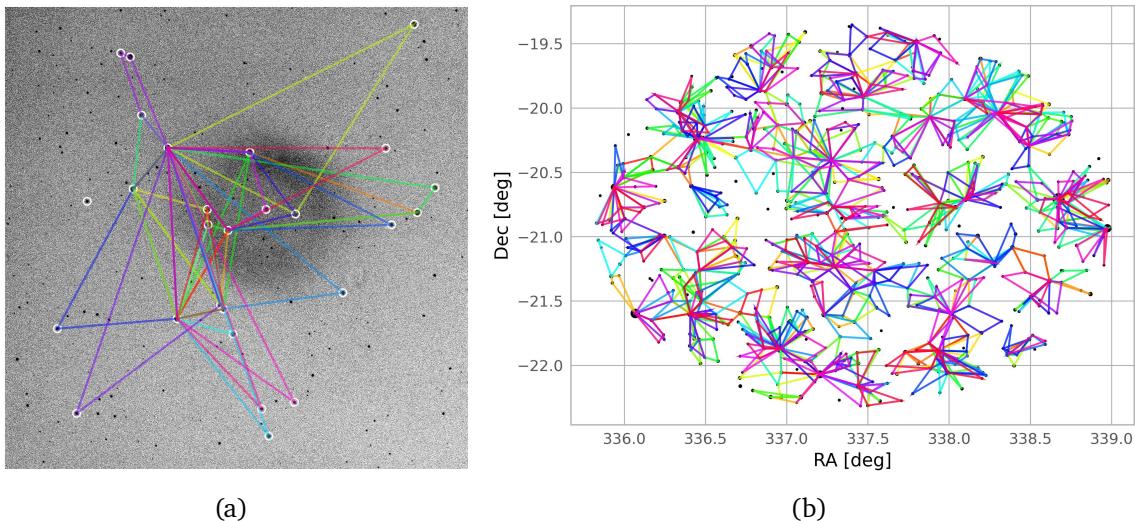
The main working example included is the plate solving of an image of the Helix Nebula, obtained with the FUT in November 2023, shown in subfigure (a) of figure 2. The choice of image is somewhat arbitrary, but by applying the algorithm to an image containing a nebula, it is better demonstrated that the star detection algorithm is robust even in cases with non-stellar image components (in this case nebulosity).



**Figure 2:** Input image of the Helix Nebula obtained with the FUT in November 2023. (a) displays the raw image input while (b) displays the detected stars in the image.

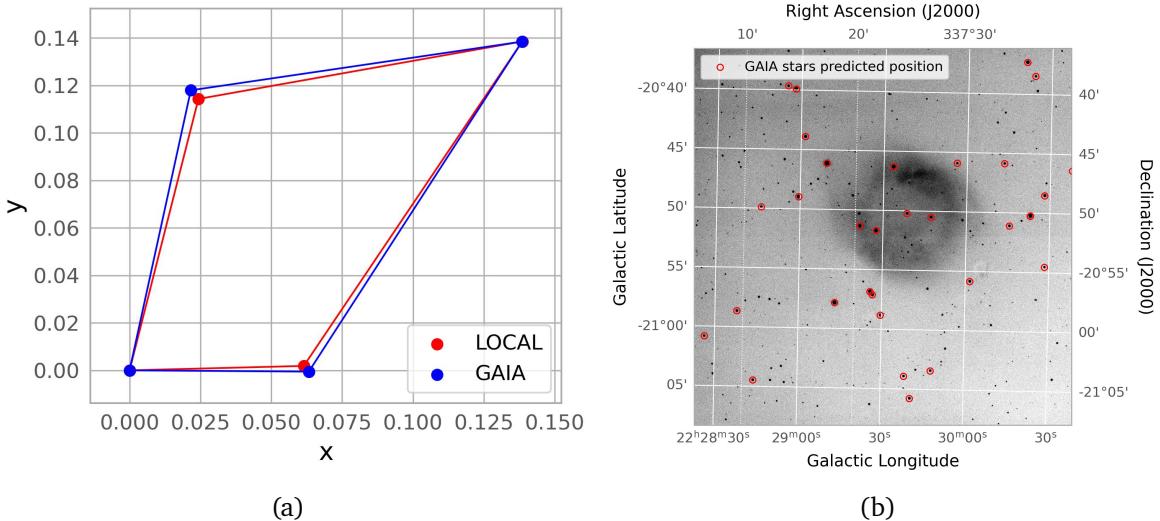
The star extraction algorithm was run on the image and the detected stars are marked in red in figure 2.

The GAIA DR2 was queried in a radius of  $2d$ , after which appropriate asterisms were built. Likewise, asterisms were built from the detected stars in the input image. These asterisms are shown side by side in figure 3.



**Figure 3:** Asterisms built from (a) the input image and (b) stars from the GAIA DR2 database.

Running the algorithm itself, a hash match was found between the GAIA index and the image asterisms with a Bayes factor  $K = 6.88 \times 10^6$ , i.e. above the threshold for a statistically significant match. The matching GAIA and image asterism is shown in subfigure (a) of figure 4, showing a great geometric similarity in the two polygons.



**Figure 4:** (a): Best-match image and GAIA hashes shown in the rotationally invariant  $x, y$  coordinate system. (b): Input image transformed into equatorial and galactic coordinates according to the newly acquired WCS. Predicted GAIA star positions are shown in red to demonstrate predictive power of the chosen hypothesis, i.e. hash match.

Finally, the match was turned into a valid WCS solution according to the FITS standard and the field, transformed to physical celestial coordinates, is shown in subfigure (b) of figure 4.

## 4 Discussion

While the algorithm seems to perform well on the small set of images considered in the debugging of the project, further work could include a more rigorous analysis of the codes ability to recognize fields from all across the sky. This could be done by programmatically querying the algorithm with images from another sky survey, e.g. the Sloan Digital Sky Survey (SDSS) (Margony, 1999) as done by Lang et al., 2010.

Further, the algorithm is relatively useless for a broader audience with its current parameters being set to only fit those of the FUT. Further work could include a more adaptive approach where all parameters are set from the image metadata.

Finally, larger scale plate-solving algorithms are typically not written in Python for performance reasons. The Helix Nebula field solved in approximately 7 seconds on a 2022 generation Macbook Air, which is sufficiently fast the scope of this project, but is likely unacceptable for larger amounts of data. For the sake simplicity, however, Python is a good choice for an exploratory project like this.

## 5 Conclusion

An algorithm for plate solving astronomical images was introduced and implemented in Python. The algorithm was showcased running on imagery of the Helix Nebula obtained by the FUT in November 2023. The results indicate a statistically significant solution was indeed found and visual inspection of the solved image confirms this. Finally, limitations of the current implementation of the algorithm are discussed.

## References

- Brown, A. G. A. et al. (Aug. 2018). “Gaia Data Release 2: Summary of the contents and survey properties”. In: *Astronomy and Astrophysics* 616, A1. ISSN: 1432-0746. DOI: 10.1051/0004-6361/201833051. URL: <http://dx.doi.org/10.1051/0004-6361/201833051>.
- Lang, Dustin et al. (Mar. 2010). “ASTROMETRY.NET: BLIND ASTROMETRIC CALIBRATION OF ARBITRARY ASTRONOMICAL IMAGES”. In: *The Astronomical Journal* 139.5, 1782–1800. ISSN: 1538-3881. DOI: 10.1088/0004-6256/139/5/1782. URL: <http://dx.doi.org/10.1088/0004-6256/139/5/1782>.
- Margony, Bruce (Jan. 1999). “The Sloan Digital Sky Survey”. In: *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 357.1750. Ed. by G. P. Efstathiou et al., 93–103. ISSN: 1471-2962. DOI: 10.1098/rsta.1999.0316. URL: <http://dx.doi.org/10.1098/rsta.1999.0316>.
- Robitaille, Thomas P. et al. (Sept. 2013). “Astropy: A community Python package for astronomy”. In: *Astronomy and Astrophysics* 558, A33. ISSN: 1432-0746. DOI: 10.1051/0004-6361/201322068. URL: <http://dx.doi.org/10.1051/0004-6361/201322068>.
- Wells, D. C., E. W. Greisen, and R. H. Harten (June 1981). “FITS - a Flexible Image Transport System”. In: 44, p. 363.