



Full length article

Calibration and testing strategies to correct atmospheric effects on star tracking algorithms

L. Jannin^a, L. Felicetti^{b,*}^a ISAE-SUPAERO, Toulouse, France^b Cranfield University, Cranfield, United-Kingdom

ARTICLE INFO

Dataset link: <https://doi.org/10.17862/cranfield.rd.24912912.v1>

Index terms:

Star tracker

Lost-in-space algorithms

Multi-poles algorithm

Polestar algorithm

ABSTRACT

Star trackers are usually considered to be the most accurate sensors, able to achieve a sub-arcminute precision. Star tracker algorithms are often tested and validated with simulated space views. Testing the algorithms with real space images is expensive as it requires implementing them on existing in-space star trackers, or to launch new satellites. This study shows that those algorithms are usually performing poorly with ground-based sky pictures and that some adaptations are necessary to take into account the atmospheric effects. The adaptation of star tracking algorithms to ground pictures could ease the prototyping phase for new star trackers, or for ground-based and air-borne star trackers, without the need to buy specific testing simulators. In order to tackle this issue, this study will start by implementing and testing two published Lost-In-Space algorithms with a simulated sensor to compare their performance against various noise sources. After comparing the space-based generated views with ground-based images, an adaptation for the aforementioned algorithms is proposed. In order to counter the effect of atmospheric extinction, the number of stars visible in the image is increased by modifying the field-of-view of the camera, the exposure time and estimating the experimental inter-star angular distance error. The idea is to match the star density used in the state-of-the-art algorithms in the experimental pictures. The modified algorithms are tested with the experimental images, and the adaptation process is validated with a good success rate.

1. Introduction

Attitude control is a necessary subsystem enabling the orientation of satellites' faces toward desired positions. Star trackers are a category of existing sensors for attitude determination and are able to provide the most accurate measurements of the attitude (Toloei et al., 2020).

The recent development of the small satellite industry creates a need for miniaturized sensors, able to be integrated into low-power and mass spacecraft. In parallel, new star identification algorithms have been developed to increase the accuracy and reliability of star trackers, as well as to increase their update frequency.

As on-ground testing and qualification of new algorithms and components is usually expensive, a simulated test is often used for developing and assessing the performance of new state-of-the-art algorithms (Schiattarella et al., 2017; Silani and Lovera, 2006; Wei et al., 2019). On the other hand, the difference in observation conditions and hardware between space and ground is severely constraining the requirements of the star identification algorithms, if used on ground-based images. Because of the poor performance of the algorithms with those pictures, some adaptations are necessary to take into account the atmospheric effects and enable the ground testing of such algorithms.

The objective of this study is to survey and quantify those differences, and propose a testing procedure to adapt two state-of-the-art algorithms for star identification with experimental ground-based images taken with a commercial camera. The motivation for such study is two-fold. First, the proposed methodology enables the fast testing of star tracking algorithm prototypes with a full acquisition pipeline, without the need to buy specific testing hardware like star tracker optical stimulator (Samaan et al., 2011; Nuno et al., 2017a). Additionally, low-end state-of-the-art optical stimulators have been recently developed with commercial off-the-shelf components (Pedrotty, 2022; Pedrotty et al., 2020; Nuno et al., 2017b; Nardino et al., 2019). Moreover, the same procedure can be applied to many ground-based or air-borne applications that require an accurate star tracking, with low-cost hardware. Possible applications are precise calibration of backyard telescopes (Riesing et al., 2018), low-cost attitude sensor for atmospheric balloon (Sreejith et al., 2014) or precise pointing of optical instruments used for object surveillance and space awareness from ground (Cordelli et al., 2019; Liu et al., 2020).

The Lost-In-Space (LIS) algorithms are a class of star tracker algorithms aiming to identify stars within the field-of-view (FOV) of a

* Corresponding author.

E-mail addresses: louis.jannin@student.isae-supaero.fr (L. Jannin), Leonard.Felicetti@cranfield.ac.uk (L. Felicetti).

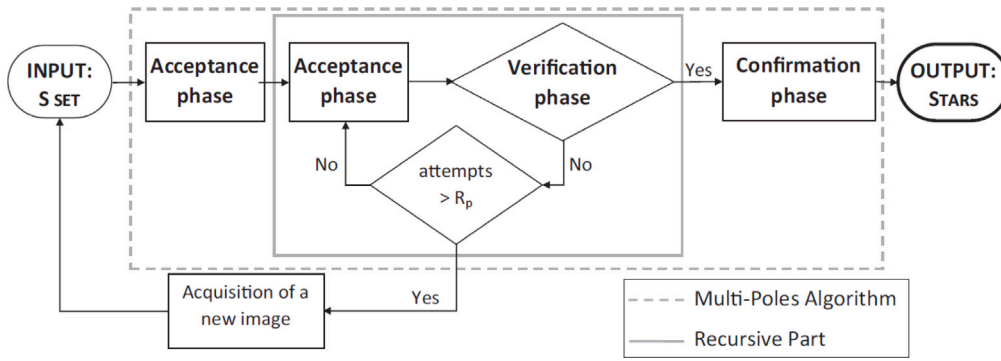


Fig. 1. General flow-chart of the Multi-Poles Algorithm, as presented in Schiattarella et al. (2017).

camera to compute the attitude of the satellite. This mode is designed to perform without prior knowledge about the attitude of the satellite. The other way of computing attitude is known as the recursive mode, where the camera tracks already identified stars in the FOV (Schiattarella et al., 2017; Spratling and Mortari, 2009; Rijlaarsdam et al., 2020). According to the work of Spratling and Mortari (2009) and Rijlaarsdam et al. (2020), the LIS algorithms are separated into two families: the *sub-graph isomorphism* algorithms where the stars are seen as vertices of a graph, and the *pattern* ones where each star is associated with a rather unique identifier that is constructed from nearby stars.

The original sub-graph isomorphism algorithm is the triangle algorithm by Liebe (1992). This algorithm is a breakthrough compared to the previous star tracking algorithms in the sense that no a-priori attitude information is required. Although this triangle algorithm exploits inter-star distances to construct polygons, the database structure resembles a pattern algorithm, making its classification quite ambiguous. Various improvements to the initial triangle algorithm have been made, with the spherical triangles (Cole and Crassidis, 2004), the planar triangles (Cole and Crassidis, 2006), and the pyramid algorithm (Mortari et al., 2004). Recent sub-graph isomorphism algorithms are for example the Multi-Pole Algorithm (MPA) (Schiattarella et al., 2017) and the geometric voting algorithm (Kolomenkin et al., 2008). Those methods are known for their reliability toward false stars noise, but the corresponding databases are massive because every pair of stars up to a given distance needs to be stored (Wei et al., 2019).

Although it extracts sub-graph features, Liebe's triangle algorithm could be considered to be also the original pattern algorithm. However, Padgett et al. (1997)'s grid algorithm is often considered to be the first, where star positions are projected into a grid. The sensor grids are then compared to the grid database and candidate stars are found for each sensor star. The difficulty of this algorithm is that the sensor grids must be aligned correctly to be matched with the grid database. Improvements of the original grid algorithm have been made again in Clouse and Padgett (2000), and furthermore with the modified grid algorithms (Lee and Bang, 2007; Na et al., 2009). The grid algorithms showed a great identification rate thanks to smaller databases, but the determination of close neighbours remained difficult in some cases. In 2006, the Polestar algorithm (Silani and Lovera, 2006) created a pattern algorithm based on radial features instead of grid features, to overcome the drawbacks of the grid algorithm. This Polestar algorithm is then extended with the radial and cyclic algorithm (Wei et al., 2019), adding a new cyclic identification feature. Other pattern algorithms involve the triangle voting (Fan and Zhong, 2018), the singular value decomposition (Juang et al., 2003) or the log-polar transform algorithm (Wei et al., 2009).

One candidate for each family of algorithms is implemented and tested in this study. Because of their contribution to the state-of-the-art of their respective category, the Multi-Poles Algorithm (MPA) (Schiattarella et al., 2017) and the Polestar Algorithm (PsA) (Silani and

Lovera, 2006) are selected. Their respective performance toward various sources of noise is tested with simulated images. Then, the differences between those simulated space views and ground-based experimental pictures are surveyed, to highlight the atmospheric effects upon the images. Finally, the two algorithms are adapted to match their original publication's star distribution by modifying the star catalogue and their internal parameters. With those adaptations, the algorithms can be tested directly with ground-based images, while maintaining a success rate of at least 80%.

2. Implementation of Lost-In-Space algorithms

In order to assess the atmospheric effects on the performance of LIS algorithms, two state-of-the-art algorithms are implemented and tested with generated space views. A brief overview of both algorithms is provided in the next sections and their internal parameters are introduced. Such parameters will then be modified during the adaptation procedure to try to match the theoretical performance with the experimental images.

2.1. Overview of the Multi-Poles Algorithm

The Multi-Poles Algorithm (MPA) is a *sub-graph isomorphism* algorithm published by in Schiattarella et al. (2017). It is especially resistant to false stars, designed to withstand a high number of Single Event Upsets (SEU) in the image, mainly generated by cosmic radiations in space. The algorithm runs over 3 main phases, as illustrated in Fig. 1.

2.1.1. Acceptance phase

The acceptance phase is repeated multiple times during the execution of the algorithm. It matches candidate stars from the on-board catalogue with various bright points detected in the image. Each iteration of the acceptance phase is made of the 4 following steps, and the corresponding illustration is shown in (a_1) and (a_2) from Fig. 2, for 2 different pole stars.

- Selection of a star in the vicinity of the centre of the image, referred to as the central star (or pole star). The other stars are listed as neighbours. In illustration (a_1) , the pole star is labelled $m^{(1)}$, and the neighbouring stars are labelled $n_i^{(1)}$, for $i \in [2; 8]$. In illustration (a_2) , the pole star is labelled $m^{(2)}$. The subscript is the neighbour numbering system, and the superscript is the corresponding pole star.
- Computation of the angular distance θ_{ij} between the central star and each neighbouring star using (1), where $\vec{s}_i$ and $\vec{s}_j$ are the vectors pointing toward stars i and j . In most catalogue the star vectors are unit vectors, but the norms $\|\vec{s}_i\|$ and $\|\vec{s}_j\|$ have been added for the general angular distance definition used by both algorithms.

$$\theta_{ij} = \arccos \left(\frac{\vec{s}_i \cdot \vec{s}_j}{\|\vec{s}_i\| \|\vec{s}_j\|} \right) \quad (1)$$

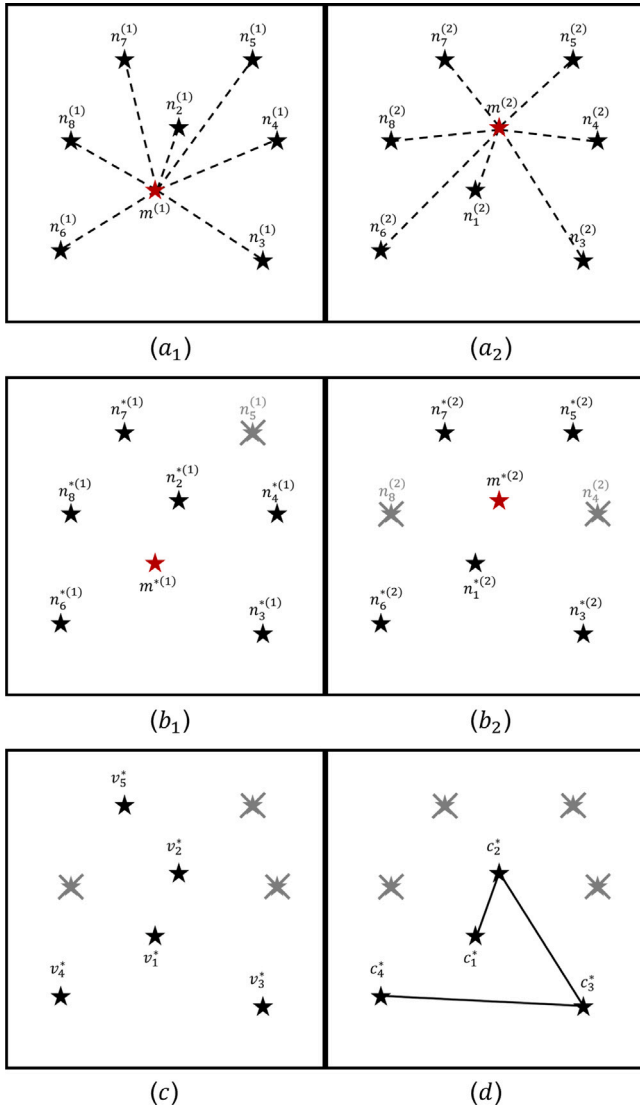


Fig. 2. Illustration of the phases of the Multi-Poles Algorithm.
Source: Adapted from Schiattarella et al. (2017)

- For each neighbouring star, all the catalogue star pairs that are separated by similar angular distances are listed as possible pairs. A user-defined tolerance ϵ_π allows to adjust the number of detected pairs.
- The most detected catalogue star within the possible pairs is accepted as the pole star. Then, all the catalogue stars paired with the pole star are accepted as neighbouring stars. For example it is possible to see that in illustration (b₁) from Fig. 2, the neighbour $n_5^{(1)}$ is not accepted by pole star $m^{(1)}$, and in (b₂) from the same figure, the neighbours $n_4^{(2)}$ and $n_8^{(2)}$ are not accepted by pole star $m^{(2)}$. The accepted stars are marked with a * superscript in the example.

2.1.2. Verification phase

Once at least two pole stars have been identified, from two different acceptance phases, the verification checks if it exists two accepted sets of stars where the pole stars are sharing the same pair. If and only if this condition is verified, the intersection of the accepted stars is computed. In the example given in (b₁) and (b₂) from Fig. 2, the pair ($m^{(1)}$, $n_2^{(1)}$) from acceptance phase (1) and the pair ($m^{(2)}$, $n_1^{(2)}$) from acceptance phase (2) are supposed to be the same catalogue pair, so

the intersection is computed and illustrated in (c) from the same figure. There are 5 stars verified, labelled v_i^* for $i \in [1, 5]$. If the catalogue pairs were different, another acceptance phase with a new pole star would be necessary.

If the number of stars identified in the intersection is more than a threshold th^* , the verification phase is successful. In this case, the set of verified stars is passed over to the confirmation phase. If not, the verification fails, and other acceptance phases are repeated until a maximum number of pole stars R_p have been tested.

2.1.3. Confirmation phase

The confirmation phase is designed to suppress any spurious match that could have made it through the verification process. This can happen if a SEU is sufficiently close to an existing catalogue star to be paired with the pole star. The verified stars from the verification phase are linked by consecutive pairs to create a chain, as illustrated in (d) from Fig. 2. It is possible to see a chain, formed from the consecutive pairs (c_1^* , c_2^*), (c_2^* , c_3^*), (c_3^* , c_4^*). It means that each star is confirmed by its predecessor and its successor in the chain. The first star of the chain, confirmed only once as it has no predecessor, is the pole star, as this star has already been confirmed by a pair of pole stars in the verification phase. Therefore, the pole star guess has usually a better guarantee to be true than the other stars in the image. The final star is linked to the first star to close the chain. In the example above, the stars from the verification phase are illustrated in (c) from Fig. 2, and it is possible to see that the star v_5^* has not been confirmed in (d). The final chain of star is labelled c_i^* for $i \in [1, 4]$. The confirmed stars are then the output of the MPA. In case no star is confirmed, the algorithm fails, and a new image has to be taken.

2.2. Implementation of the Multi-Poles Algorithm

The MPA has been implemented with the following parameters, with respect to the original publication's choices for the symbols (Schiattarella et al., 2017):

- **Maximum number of poles:** $R_p = 20$. This number means that after trying R_p different poles through the acceptance-verification loop, the algorithm will stop.
- **Verification threshold:** $th^* = 5$. This is the minimum number of stars that need to be verified between two poles during the verification phase for this step to be considered a success.
- **Angular distance tolerance:** $\epsilon_\pi = 1 \text{ arcmin}$. This is the maximum variation in angular distance $\theta_{ij,cat}$ between two catalogued stars s_i , s_j and the corresponding sensor distance $\theta_{ij,sensor}$ such that (s_i , s_j) is accepted as a candidate pair:

$$|\theta_{ij,cat} - \theta_{ij,sensor}| \leq \epsilon_\pi \quad (2)$$

2.3. Overview of the Polestar Algorithm

The Polestar Algorithm (PsA) is a *pattern* algorithm published in Silani and Lovera (2006). It significantly improves the precedent pattern algorithms with a novel identification technique based on radial features. In opposition to the sub-graph isomorphism algorithms, the information stored in the on-board star catalogue is not only distances between couples of stars but includes also star patterns, hence the name of this category. The pattern generation technique creates a rather unique pattern for each star, enabling the identification of the stars by finding the same pattern in an image.

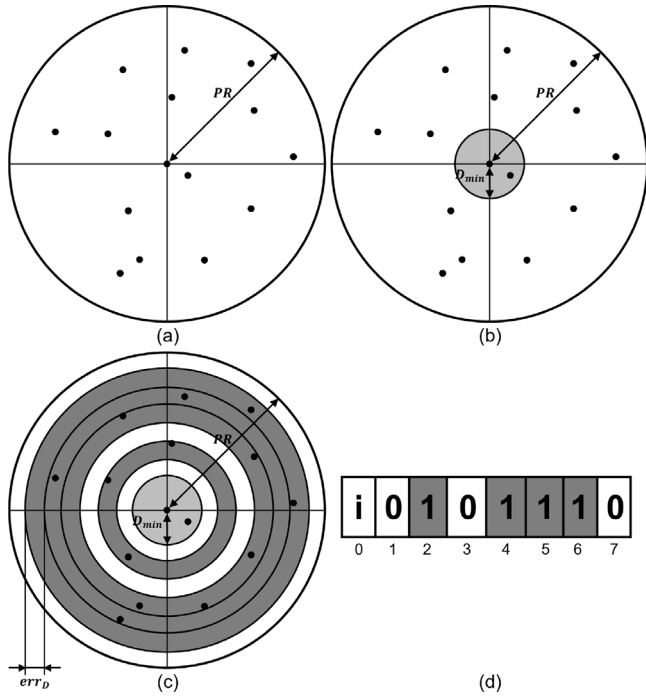


Fig. 3. Illustration of the generation of the bar code for a pole star.
Source: Adapted from Silani and Lovera (2006)

2.3.1. Pattern generation

The first step of the algorithm is to generate the patterns associated with the selected bright points in the image. The brightest points on an image are often considered the most reliable. Hence, in order to filter false stars, not all image stars will be used for pattern generation. Those patterns are also generated beforehand for all the stars in the star catalogue, and stored in a lookup table.

The patterns are generated by considering the stars located within a given range of the pole star: $[D_{min}; PR]$, where D_{min} and PR are respectively the shortest distance accepted between the pole star and another star, and the Pattern Radius, as shown in (a) and (b) from Fig. 3. The range between D_{min} and PR is then divided into small concentric rings of a given length err_D , as shown in (c). Finally a bar code is generated for the pole star, with a value of 1 for the rings containing at least one star, and a value of 0 for the empty ones, as shown in (d). In the illustrated example, the first, third and last rings are empty, thus the bar code is 0101110.

In order to know efficiently if a given interval contains a star, the distances to the pole star are rounded. If θ_{ij} is the raw distance, computed with (1), the rounded angular distance $\hat{\theta}_{ij}$ is obtained by a discretization of length err_D . The length N_{max} of the bar code is then given by the number of subdivision of the $[D_{min}; PR]$ interval. This number depends on err_D and is given by (3).

$$N_{max} = \text{floor} \left(\frac{PR - D_{min}}{err_D} \right) + 1 \quad (3)$$

floor is the function that rounds a number to the nearest precedent integer.

2.3.2. Candidates generation

For each bright point in the image, a bar code is generated, and the lookup table from the star catalogue is used to find suitable candidate stars. For each 1 in the bar code of a sensor star, the corresponding row of the lookup table is accessed, and all the catalogue stars in this row are counted as possible candidates. The appearance count of each candidate is then recorded: the new candidates create a new counter,

whereas the already registered candidates' counter is incremented. Finally, the stars whose counter is at least equal to the maximum counter minus one are listed as candidates for the pole star.

2.3.3. Polygon verification

Once each sensor star is associated with a list of potential candidates from the star catalogue, the candidates are checked by pairing them in polygons, with an increasing number of edges. At first, pairs are made between candidate stars, then triangles, then 4-edges polygons, and so on until all the candidates are accepted or rejected. This final step ensure that all the stars returned by the PsA are confirmed by the other stars in the image.

2.4. Implementation of the Polestar Algorithm

The PsA has been implemented with the following parameters, with respect to the original publication's choices for the symbols (Silani and Lovera, 2006):

- **Minimal separation distance:** $D_{min} = 0.4^\circ$. This is the minimum distance allowed between the pole star and a pattern star.
- **Pattern radius:** $PR = 10^\circ$. The pattern radius is simply equal to half the FOV of the simulated camera, such that a central star on the image would have its pattern covering the entire image. This value is coherent with the one in the original publication.
- **Subdivision length:** $err_D = 2 \text{ arcmin} = 0.033^\circ$. This is the length between two consecutive radii for the bar code pattern. This is the same order of magnitude than the value 0.02° from the original publication.
- **Number of patterns:** $N_{max} = 290$. This is the length of the bar codes.
- **Minimum match:** $\beta = 10$. This is the minimum number of matches between a sensor pattern and a catalogue pattern for the star to be considered.
- **Angular distance tolerance:** $\epsilon_{pairs} = 2 \text{ arcmin}$ and $\epsilon_{triangle} = 3 \text{ arcmin}$. This is the maximum difference accepted for the distance between catalogued stars and sensor stars.

3. Testing of Lost-In-Space algorithms

In order to create the simulated views and test the algorithms, an automated generator of images is implemented. The image generator produces 100 images for each test point, and each image is generated with a random orientation in space. This is because the star distribution along orientation is not uniform, and it can affect the performance of the algorithm. The simulated camera used for the computation of the views is a rectilinear camera with a square FOV of 20° and a resolution of 1024 by 1024 pixels. Its light sensitivity is such that it is able to detect stars up to a magnitude of 6, and the camera distortion is neglected at first. The Hipparcos (HIP) catalogue from the European Space Agency will be used as the reference for star positions (Perryman, 1997).

In this section, two different analyses are done to assess the performance of the algorithms. At first, the reference performance are computed without any source of noise, in order to provide a benchmark for the comparison of the success rate of the algorithms with the experimental setup. Then, the robustness of the algorithms toward various sources of noise is assessed and quantified. This robustness will help to determine which parameters are critical for the adaptation of the algorithms in the experimental tests.

3.1. Reference performance

The reference performance of the algorithms are obtained with simulated images and without any source of noise. The stars identified by the algorithms are compared to the reference list of stars in the FOV and 3 metrics are monitored in the tests:

Table 1
Reference performance of the algorithms.

Algorithm	Identification rate	Success rate	Elapsed time
Multi-poles	100%	99%	0.5 s
Polestar	100%	100%	0.6 s

- *Identification rate*: this number represents the percentage of images for which the algorithm is able to identify at least 3 stars. It is important to note that an algorithm can possibly return a list of stars, but this list can contain false positives.
- *Success rate*: this number represents the percentage of images that have been successfully identified, and for which the algorithm's guess is true for each star.
- *Elapsed time*: this is the average time needed for the algorithm to run through one image. For the elapsed time computation, the algorithms are tested with an AMD Ryzen 3 3200U 2.6 GHz processor and 6 GB of RAM capacity.

In most cases, the identification rate and the success rate differ by only a few percent, mostly due to multiple star systems that have multiple entries in the HIP catalogue. The algorithm is then able to correctly identify the star system, but when the inter-star distances is smaller than the algorithm's threshold, a wrong star number can be outputted. The reference performance of the MPA and the PsA algorithms are given in Table 1.

3.2. Robustness toward sources of noise

In order to assess the performance of the chosen algorithms toward various sources of noise, simulated space views are generated and fed to the MPA and the PsA algorithms. The results of those tests will be compared to the experimental values of noise observed in the ground-based images, to identify the parameters that would require an adaptation. The algorithms are tested toward false stars, position noise and magnitude noise, as described below.

- *False stars*: adding bright points on the image that are not registered in the catalogue. Those false stars could be caused by SEU, or other objects such as dusts, planets of satellites that might be captured in the FOV of the camera. To simulate this effect, a fraction of false stars (from 0 to 140%) with a random magnitude ranging from -1 to 6 as well as a random position on the image, is added.
- *Position noise*: the centroiding techniques used to precisely locate star positions on the image are limited by the light acquisition of the camera, image distortion and resolution. As such, the star positions on the image can be slightly shifted compared to the catalogue positions. This effect is simulated by shifting the position of the stars on the image by a given number of pixels (from 0 to 2.5), in a random direction.
- *Magnitude noise*: the detected magnitude of the stars can be modified by the dispersion of the light on the different pixels of a single light source and by the settings of the camera. In addition to that, some stars can have a variable magnitude. It means that some stars that should be detected by the camera are actually invisible on the image, and some stars that should be too faint to be captured are visible on the image nonetheless. To simulate this effect, the magnitude of the stars on the image is randomly increased or decreased, following a Gaussian distribution with a standard deviation ranging from 0 to 1 depending on the desired level of noise. The stars are then filtered so that only the ones that are bright enough are captured on the image.

The result for the tests toward the various sources of noise are available in Fig. 4, and explained below.

Table 2
Camera parameters for various implementation of LIS algorithms.

Source	Resolution (Mpx)	FOV (°)	Maximum magnitude
Schiattarella et al. (2017)	1	20	6.5
Silani and Lovera (2006)	0.1	7.5×10	7
Wei et al. (2019)	1	12	6.5
Fan and Zhong (2018)	4	15	6

3.2.1. Multi-Poles Algorithm

It is possible to see that the MPA is able to withstand noise levels of 75% false stars, 1.5 pixels of positional noise and uncertainties of 0.65 for the magnitude while keeping a success rate above 80%. Given the camera resolution of 1024 pixels and the simulated FOV of 20° , this represents a shift in star position of 1.76 arcmin. It is the same order of magnitude than the 1.32 arcmin found in the testing of the multi-poles algorithm done in Fan and Zhong (2018), or the 1.125 arcmin for the tests in Wei et al. (2019). The performance against magnitude noise is also tested, with a success rate of 93% for a level of 0.4. Again, this is close to the 95% success rate for the same noise level in Wei et al. (2019).

3.2.2. Polestar algorithm

The PsA is able to maintain a success rate of 80% with up to 80% false stars, 1.25 pixels (1.46 arcmin) of positional noise and uncertainties of 0.7 for the magnitude. In the original publication, the position noise is applied following a Gaussian distribution, and the 80% success rate is reached for a standard deviation σ_p of about 0.28 pixels, which corresponds to 1.31 arcmin for $3\sigma_p$. The PsA is also tested in Fan and Zhong (2018), and is reported to withstand a shift in star position of 3 pixels for a success rate of 80%, where a simulated FOV of 14.7° with 2048 by 2048 pixels is used. This is equivalent to 1.29 arcmin of shift in star position, again coherent with the aforementioned values. The performance of the PsA is similar to the ones of the MPA, toward every source of noise.

4. Adaptation of the experimental setup

In order to match the star distribution used in the simulated tests, a survey of the experimental camera settings used in the testing of the aforementioned algorithms is drawn. Although star trackers differ in many ways depending on their respective mission, on the targeted accuracy and on the operating environment, there are some common features shared by many star identification pipelines, that are described below. Star trackers usually have a small FOV, a low pixel count and a high sensitivity, as explained below. The implementation of LIS algorithms done in Schiattarella et al. (2017), Silani and Lovera (2006), Wei et al. (2019) and Fan and Zhong (2018) is surveyed to find the camera parameters used in the simulated tests. The results are presented in Table 2, and described hereafter.

Typically a few megapixels (Mpx) are enough for the star identification. A higher resolution would mean a bigger camera, which could not be mounted on the satellite. It would also mean heavier pictures, which could be difficult to manage with a limited computational capability. Ultimately, heavy images would deteriorate the LIS algorithms identification rate due to an excessive duration for image processing.

The typical star tracking camera's FOV is around 15° wide. This is enough to capture a sufficient number of stars for the identification algorithm, given the sensitivity. Wider cameras would require wider lenses that take more space and weight on the satellite, and it also increases the optical distortion of the image.

As the satellite is in space, it has no atmospheric distortion or filtering, nor light pollution. Usually, stars with magnitude 6 or above are able to be captured by the camera.

The application of these simulated camera settings to experimental ground-based images often produces a degradation of the algorithms'

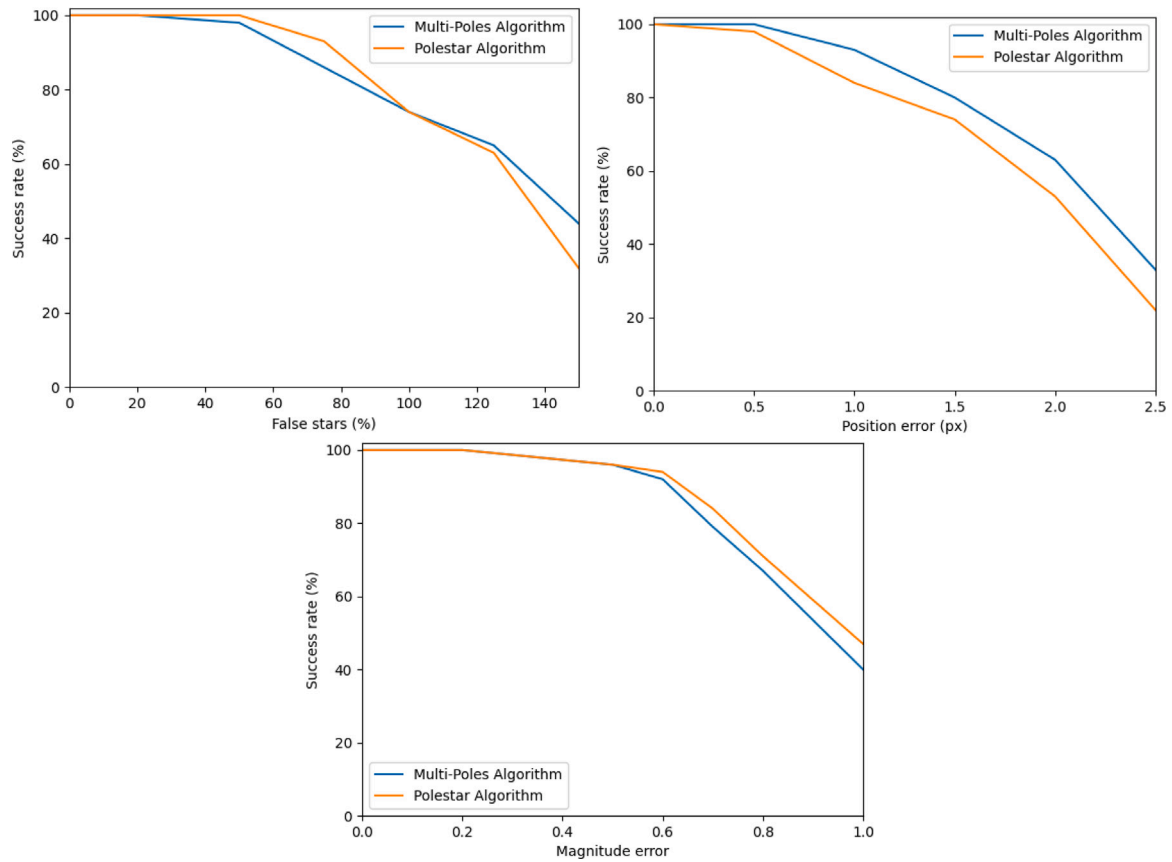


Fig. 4. Performance comparison of the Multi-Poles Algorithm and the Polestar Algorithm toward false stars (top figure), position noise (central figure) and magnitude noise (bottom figure).

performance which inhibits any proper testing of the algorithms on ground. Therefore, the effect of the atmosphere on the experimental images needs to be considered. The next sections aim at describing and quantifying those effects so that an efficient adaptation procedure can be proposed to match the star distributions between simulated images and ground-based images.

4.1. Atmospheric effects

The atmosphere has a filtering and scattering effect on the light received from space. Those effects are known as atmospheric extinction and atmospheric blurring (or seeing) (Schaefer, 1986), and are monitored by space observatories (Patat et al., 2011; Rufener, 1986; Burki et al., 1995).

Atmospheric extinction is the absorption of starlight by the Earth's atmosphere. This absorption is mainly due to Rayleigh scattering by air molecules, to general scattering by fine particles and to molecular absorption by Oxygen and Ozone. This effect is stronger as the airmass above the observer increases. Hence, the level of extinction is lowest around the zenith and highest around the horizon (Schaefer, 1986).

The atmospheric blurring arises from the various refraction indexes of the atmospheric layers. Those layers have slightly different physical parameters that modify the refraction index, bending the starlight by a different angle depending original position of the stars. This effect is further strengthened by the atmospheric motion, that dynamically changes the refraction index over time.

4.2. Adaptation of the experimental setup

The star distribution visible with the simulated camera with a FOV of 20° and a maximum magnitude of 6 is surveyed, as it needs to be

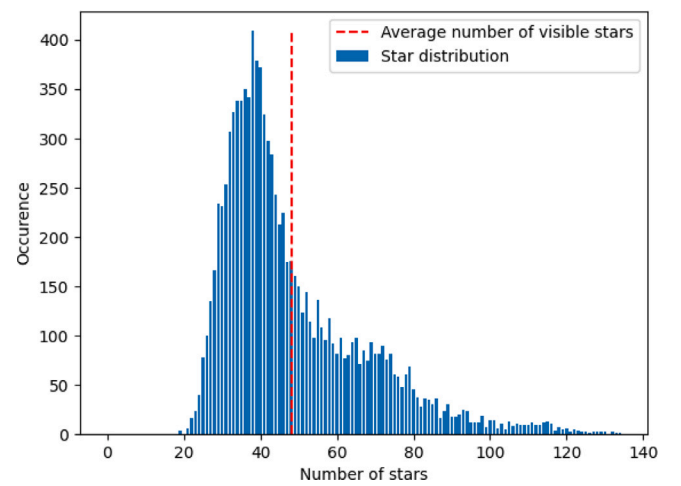


Fig. 5. Number of stars up to magnitude 6 visible in a 20° field-of-view.

matched by the experimental setup. To have a confident result, 10,000 random orientations are generated and the corresponding distribution for the number of visible stars is reported in Fig. 5. It is possible to see a peak around 40 stars per image with almost 400 occurrences, and an average of 48 visible stars among all the cases.

With zenith atmospheric extinction and suburban light pollution levels due to the atmosphere, the maximum observable magnitude is reduced to 4. The same analysis is repeated with the same FOV of 20° and a limiting magnitude of 4. The results are reported in Fig. 6. It appears that the average number of visible stars is 5, with a peak at 4,

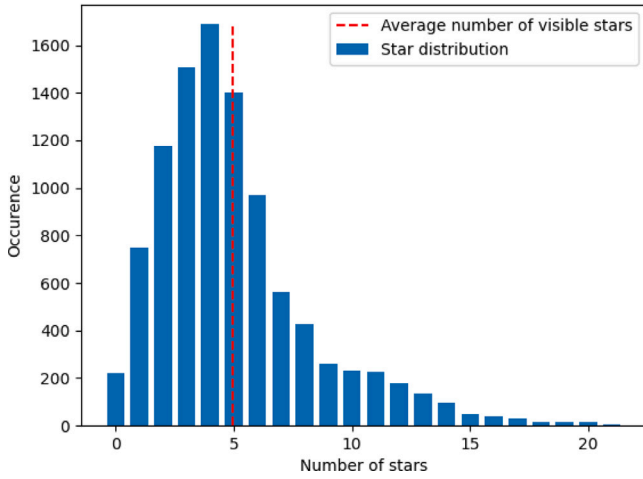


Fig. 6. Number of stars up to magnitude 4 visible in a 20° FOV.

with an occurrence above 1600. In 10% of the orientations, the number of visible stars is less or equal than 1, which makes it impossible to identify the star nor compute the attitude.

In order to get closer to the typical in-space star distribution, the FOV, resolution and sensitivity of the experimental camera need to be adapted. A wider FOV for the camera is necessary, as well as a longer exposure time to capture more stars on the image. A wider FOV implies to increase the resolution in order to keep a sufficiently low arcminute per pixel rate. In the experimental setup, this quantity is around 1.17 arcminute per pixel. The experimental setup used in this study is a SM-G960f rear camera, with a square FOV of about 51°, a resolution of 3024 by 3024 pixels and a variable exposure time. This new resolution means that a pixel corresponds to 1.01 arcminute. This model of camera is illustrated in Fig. 7. Additionally, a diagram of the experimental setup is given in Fig. 8. The raw image from the camera is transferred to a star detection pipeline that corrects the image distortion and identifies the bright points. This list of potential stars is then given to the selected LIS algorithm to identify at least 5 stars from the star catalogue. Finally, with the catalogue coordinates of the identified stars, the camera's attitude can be precisely derived with the QUEST algorithm (Shuster, 1978, 1981).

The default exposure time of the camera is 0.2 s, which is barely enough to capture stars up to magnitude 3. This is not enough, and after several tests, the exposure time is set to 10 s, so that stars up to magnitude 5 can be captured. The expected distribution for the number of stars visible is given in Fig. 9. The average number of visible stars is 96, with a minimum number of visible stars per image above 50. This distribution involves a lot more stars than the one of the simulated views, which is due to the larger FOV.

Although the modification of the FOV and the exposure time allowed to drastically increase the number of visible stars on the image, the atmospheric blurring is still increasing the inter-star angular error as well as the image distortion. In order to reduce the intrinsic distortion of the image, the camera parameters are estimated with the chessboard method (Hartley and Zisserman, 2003) so that the Python package OpenCV (OpenCV, 2014) can be used to undistort the images afterward. After this step, the experimental shift for star positions observed on the image is reduced to 5 pixels. With a 51° FOV and a resolution of 3024 pixels, the angular accuracy of a pixel is roughly 1.01 arcminute. It means that the inter-star distance can have an error of a few arcminutes, which is higher than the value of ϵ_π used in the definition of the MPA with the simulated views. It is also higher than the maximum level of shift in star position used in the simulated tests. The change in ϵ_π has a direct effect on the number of pairs found in the

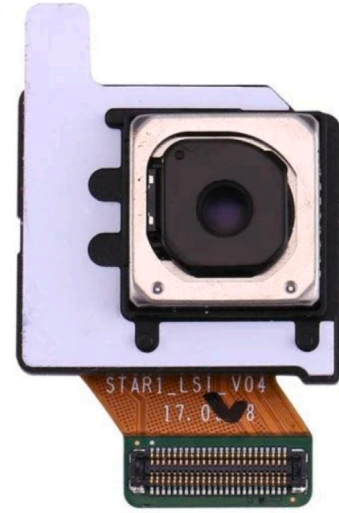


Fig. 7. Picture of a SM-G960f rear camera.

star catalogue, thus increasing ϵ_π will increase the spurious correlations for a given inter-star distance.

In order to monitor the raw performance of the Polestar and the MPA, 35 experimental pictures are taken with this setup. Those pictures are spanning the night sky visible in the Northern hemisphere during the months of June to August 2022. The atmospheric conditions are as much as possible comparable for all pictures, and cloudy pictures were not included. An experimental picture is given as an example in Fig. 10. The gray scale was inverted to match the paper colour. The dataset is accessible in an open-access repository (Jannin and Felicetti, 2023), with the camera calibration parameters, in order to reproduce the results.

The only modification done in the implementation of the algorithms is the change of catalogue to match the maximum magnitude of 5, as well as the inter-star error set to 10 arcminutes. The catalogue was not restricted to stars visible in the Northern hemisphere, in order to be able to compare the experimental pictures with the simulated tests. As the star position can be shifted up to 5 arcminutes, then the inter-star distance error is up to twice the shift in position, hence the 10 arcminutes used in the definition of the algorithms. In order to compare the average performance on the simulated results with the average performance on the experimental pictures, the aforementioned implementation is used. This results in an average success rate on the experimental pictures of 30% for the MPA, and 10% for the PsA, which is far below the average success rate of 99% for the simulated images. This shows that the algorithms need to be adapted, as solely adding more stars to the image results in a poor success rate.

5. Adaptation of the algorithms

The adaptation of the algorithms aims at increasing their experimental performance, by modifying the catalogue, the number of bright stars, and their internal parameters.

The first modification relates to the star catalogue, the idea is to keep only the ξ brightest stars of an image in order to stay close to the simulated distribution. The maximum magnitude of the ξ^{th} brightest star of the image will then be estimated given the FOV of the camera and the distribution of the HIP catalogue. As the star distribution amongst increasing magnitude is nearly exponential, diminishing the maximum magnitude is greatly reducing the potential mismatches.

The next modification involves the precise estimation of the FOV of the camera, as well as the inter-star angular error given the atmospheric blurring and the residual distortion after the correction. This will help to reduce the threshold for the error ϵ_π , thus further reducing the number of spurious matches.

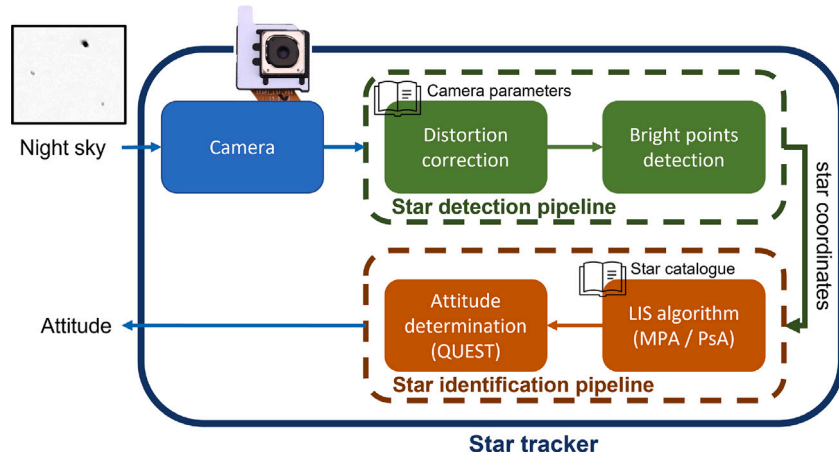


Fig. 8. Flow diagram of the star tracker pipeline.

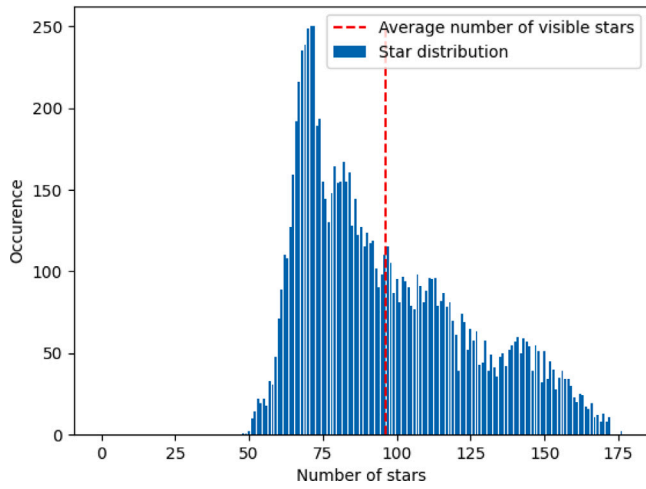


Fig. 9. Number of stars up to magnitude 5 visible in a 51° FOV.

5.1. Catalogue reduction

As explained in the testing of the LIS algorithms, the HIP catalogue may have several entries for the different stars of multiple star systems. In the context of this study, the camera may not be able to separate them on the image, given the 1.01 arcminute per pixel resolution. This can lead to wrong identification catalogue-wise for stars that appear on the same pixel. The cumulative number of stars in the HIP catalogue is illustrated in Fig. 11, where both the filtered and unfiltered catalogues are considered.

As the number of stars is plotted in logarithmic scale, it is possible to see that the star distribution among increasing magnitude follows an exponential law. The number of stars in the filtered catalogue up to magnitude 4 is 428, but increases to 1353 when the maximum magnitude is 5. Additionally, the star distribution for a FOV of 51° and a maximum magnitude of 4 is given in Fig. 12. The average number of visible stars is 31, which is sufficient to run the LIS algorithms and derive the attitude. This is closer to the simulated star distribution given in Fig. 5.

Furthermore, in 90% of the orientations, more than 20 stars are visible. It means that in more than 90% of the cases the $\xi = 20$ brightest stars have a magnitude of 4 or below. This value gives an insight into how many stars have to be fed to the LIS algorithms to work properly with the HIP catalogue. However, there are many more than 20 bright points that can be detected in the image because of the sensitivity of

the camera and the long exposure time. But if the star catalogue is restricted to a given magnitude, then all the stars above this magnitude will not be included in the catalogue, thus being considered as false stars. In order to feed only the ξ brightest stars to the LIS algorithm, the bright points detected during the image processing step of the star tracker pipeline need to be weighted by their relative intensity.

In reality, due to the atmospheric blurring, a star appears as a cluster of bright pixels, that is isolated from the background during the clustering step. The boundary of a cluster is set up by an adaptive threshold during the binarisation step (Gonzales-Barron and Butler, 2006). The centre of a cluster is computed with the centroiding method, taking into account the relative intensity of the pixels (Vyas et al., 2009). This kind of cluster is illustrated in Fig. 13, with a closeup view of the star Vega, as it appeared in one of the experimental images. The gray scale is inverted to match the colour paper, as the background is usually black. It is possible to sum the relative luminosity of each pixel of a cluster to determine the apparent brightness of a star. Those clusters are then sorted by decreasing brightness, and the ξ brightest ones are fed to the LIS algorithms.

At this point, the HIP has been modified. But the LIS algorithms still need to be adapted, and take into account the previous modifications.

5.2. Multi-Poles Algorithm

The MPA is tested with the following parameters:

- *Maximum number of poles:* $R_p = \xi = 20$. All the stars fed to the algorithms can be tested for the poles.
- *Verification threshold:* $1h^* = 5$. This value remains unchanged.
- *Angular distance tolerance:* $\epsilon_\pi = 5.5$. This parameter is obtained after some fine-tuning, as explained below.

In order to find more precisely a good guess for ϵ_π , several values are tested, ranging from 1 arcminute to 15 arcminutes. The result of this analysis is available in Fig. 14. When the inter-star distance error is too small, the performance decreases because it is impossible to find a sufficient number of matches within the star catalogue for the image's stars. On the other hand, when ϵ_π is higher than 10 arcminutes, the performance decreases because the spurious matches within the star catalogue outnumber the correct stars. Good values for ϵ_π are then ranging between 4 and 8 arcminutes, with a peak success rate of 86% for $\epsilon_\pi = 5.5$ arcminutes.

At the same time, a parallel analysis of the camera parameters is done to fine-tune the FOV. Indeed this value is given in the specification sheet of the manufacturer, but a sub-degree precision is needed here as this value determines the angular distance between points on the image. Several values are tested, and the result is shown in Fig. 15. The peak performance is obtained for a FOV of 51.35°.

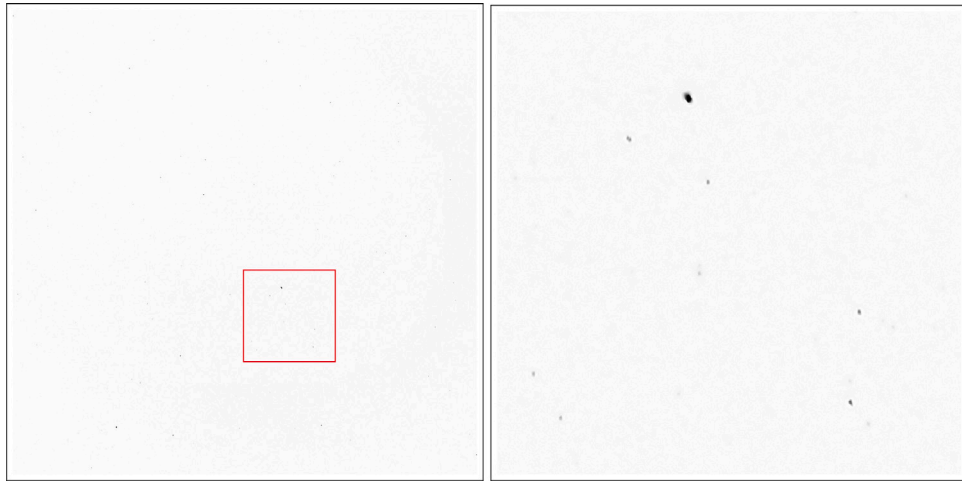


Fig. 10. Example of an experimental image (left) with a close up of the red squared area, showing the Lyra constellation (right). The gray scale of the image is inverted to match the paper colour.

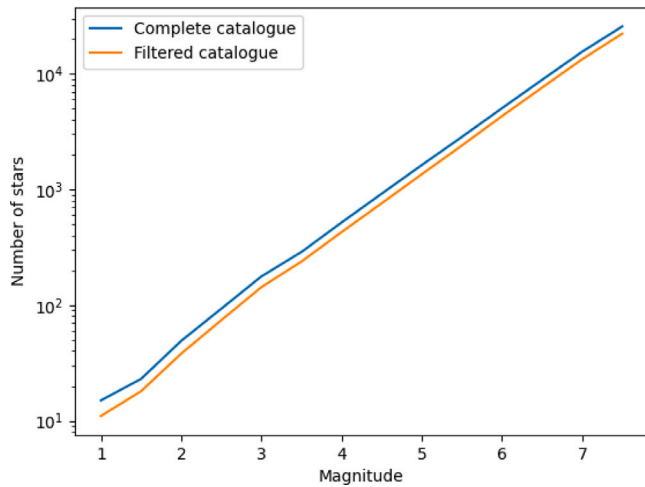


Fig. 11. Cumulative number of stars per magnitude in the Hipparcos catalogue.

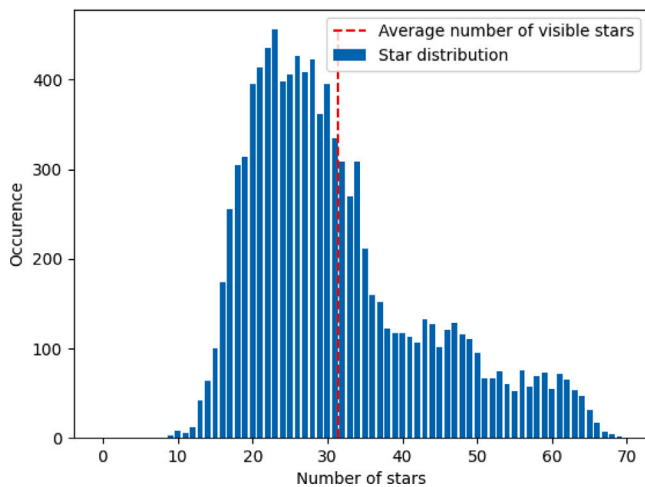


Fig. 12. Number of stars up to magnitude 4 visible in a 51° FOV.

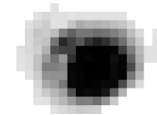


Fig. 13. Close up view of a pixel cluster for the star Vega in the Lyra constellation. The gray scale is inverted to match the paper colour.

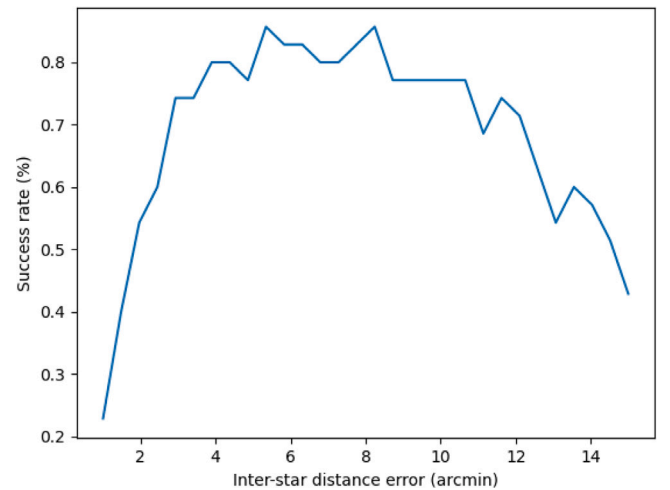


Fig. 14. Success rate for the Multi-Poles algorithm, for various values of inter-star distance error.

5.3. Polestar algorithm

The PsA has more internal parameters than the MPA, but the angular distance tolerance for the pairs of stars ϵ_{pairs} has the same meaning as ϵ_{π} so the same value of 5.5 arcminutes is used. The other important parameters are defined below, with an important change for the subdivision length err_D .

- *The number of pole stars that are used to generate the bar codes.* This number has the same behaviour as R_p in the case of the MPA, so the value of $\xi = 20$ is used, to stay coherent with the restriction of the catalogue and the star distribution.
- *Minimum match $\beta = 10$.* The same value for the number of matches between the sensor pattern and the catalogue pattern is kept.

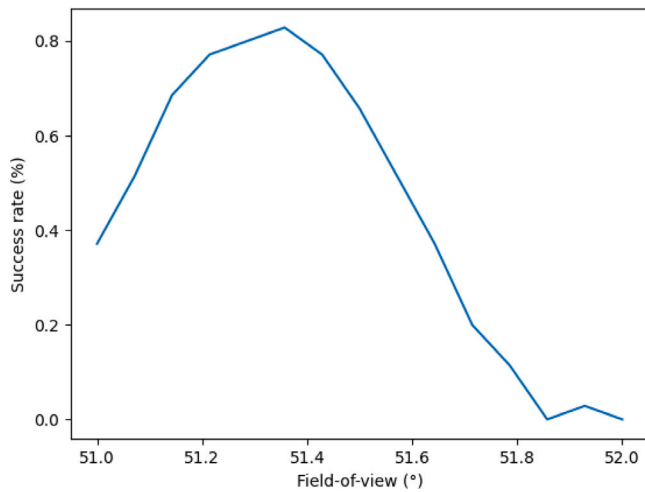


Fig. 15. Success rate for the Multi-Poles algorithm, for various values of the field-of-view of the camera.

- *Minimal separation distance* $D_{min} = \frac{2}{100} \times FOV$. This adaptation is done to keep the same ratio between D_{min} and the FOV as in the original publication (Silani and Lovera, 2006).
- *Pattern radius* $PR = \frac{1}{2} \times FOV$. Again, the same ratio is kept, with respect to the original publication.
- *Subdivision length* $err_D = 10$ arcminutes. This value is more difficult to define, as it has a direct impact on the length of the bar code, and the success rate of the algorithm. The fine-tuning of this parameter is explained below.

The length of the bar code is defined by the number of concentric circles drawn around a pole star. This number is determined by the length of the stripes, which is err_D . The test of various values of err_D is presented in Fig. 16. The performance of the Polestar algorithm stays nearly constant as long as err_D is above 10 arcminutes. This value is interesting because it is approximately twice the value of ϵ_π . As the star position is not very precise, it can be shifted toward any given direction by a fraction of ϵ_π , which means a region of $2\epsilon_\pi$ is enough to make up for the imprecision, as explained in Silani and Lovera (2006). Indeed, the best choice seems to be $err_D = 10$ arcminutes, for a success rate of 86%. Ultimately the success rate is bound to drop when the err_D is too high, because the length of the bar code N_{max} will be too small to discriminate between the catalogue stars. With $err_D = 10$ arcminutes, it means that $N_{max} = 147$, which is approximately half the value used in the simulated tests of the PsA.

6. Conclusion

This study demonstrated that LIS algorithms designed for space applications can be used with ground-based images, while maintaining a good success rate. Specific adaptations are suggested for both the capturing and processing of the images, as well as the modification of the intrinsic parameters of the LIS algorithms. Adaptation on the capture and processing of the images involves the increase in exposure time and field-of-view in order to capture more stars on the image, as well as the restriction of the catalogue to match the simulated star distribution. On the other hand, the modifications to the internal parameters of the LIS algorithms are necessary to better match the experimental inter-star distance error.

Throughout the application of the proposed methodology, it was possible to obtain an average success rate of 86% for both the MPA and the PsA algorithms, which is comparable with the performances obtained with simulated images for noise levels of 1 to 1.5 pixels of shift in star position, a magnitude uncertainty of 0.65 to 0.7, and a

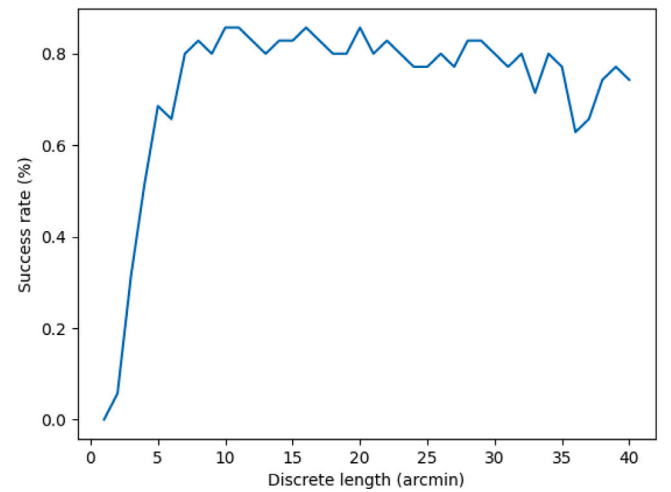


Fig. 16. Success rate for the Polestar algorithm, for various values of the discrete step err_D .

false star rate between 80% and 90%. Even though the experimental performances are slightly lower than the simulated performances, the results show that this procedure can be used to mitigate some of the atmospheric effects applied to the ground-based images, and proficiently allows for preliminary tests of LIS algorithms.

The current pipeline is not designed to withstand atmospheric perturbations like clouds that could reduce the number of visible stars under a minimum threshold of five stars. This threshold is necessary for the identification pipeline through the MPA and PsA algorithms. Improvements of such algorithm could include robustness against dynamic effects like spacecraft rotation or adverse atmospheric conditions. Such modifications would only concern the image processing pipeline involved in the star detection, but not the star identification itself with the LIS algorithms, which is the main focus of this study.

Adapting star tracking algorithms to ground images has the potential to ease the prototyping phase of emerging star trackers, including those designed for ground-based and airborne applications alongside the space-based ones, thereby reducing the need to invest in specialized testing simulators and stimulators.

CRedit authorship contribution statement

L. Jannin: Data curation, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft. **L. Felicetti:** Conceptualization, Project administration, Supervision, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The experimental image dataset is available at: <https://doi.org/10.17862/cranfield.rd.24912912.v1>.

References

- Burki, G., Rufener, F., Burnet, M., Richard, C., Blecha, A., Bratschi, P., 1995. The atmospheric extinction at the E.S.O La Silla observatory. *Astron. Astrophys. Suppl. Ser.* 112, 383–394.
- Clouse, D.S., Padgett, C.W., 2000. Small field-of-view star identification using bayesian decision theory. *IEEE Trans. Aerosp. Electron. Syst.* 36, 773–783.

- Cole, C.L., Crassidis, J.L., 2004. Fast star pattern recognition using spherical triangles. In: AIAA/AAS Astrodynamics Specialist Conference. p. 23. doi:10.2514/6.2004-5389.
- Cole, C.L., Crassidis, J.L., 2006. Fast star pattern recognition using planar triangles. *J. Guid. Control Dyn.* 39. doi:10.2514/1.13314.
- Cordelli, E., Schlatter, P., Lauber, P., Schildknecht, T., 2019. Use of a night-tracking camera for characterization and orbit improvement of defunct spacecraft. In: 1st NEO and Debris Detection Conference.
- Fan, Q., Zhong, X., 2018. A triangle voting algorithm based on double feature constraints for star sensors. *Adv. Space Res.* 61, 1132–1142.
- Gonzales-Barron, U., Butler, F., 2006. A comparison of seven thresholding techniques with the k-means clustering algorithm for measurement of bread-crumbs features by digital image analysis. *J. Food Eng.* 74, 268–278.
- Hartley, R., Zisserman, A., 2003. *Multiple View Geometry in Computer Vision*, Second Edition. Cambridge University Press.
- Jannin, L., Felicetti, L., 2023. Supporting dataset for ‘calibration and testing strategies to correct atmospheric effects on star tracking algorithms’ cranfield online research data (CORD). doi:10.17862/cranfield.rd.24912912.v1.
- Juang, J.N., Kim, H., Junkins, J., 2003. An efficient and robust singular value method for star pattern recognition and attitude determination. *Adv. Astronaut. Sci.* 18, 115–120.
- Kolomenkin, M., Pollak, S., Shimshoni, I., Lindenbaum, M., 2008. Geometric voting algorithm for star trackers. *Aerosp. Electron. Syst. IEEE Trans.* 44, 441–456. doi:10.1109/TAES.2008.4560198.
- Lee, H., Bang, H., 2007. Star pattern identification technique by modified grid algorithm. *IEEE Trans. Aerosp. Electron. Syst.* 43, 1112–1116.
- Liebe, C.C., 1992. Pattern recognition of star constellations for spacecraft applications. *IEEE Aerosp. Electron. Syst. Mag.* 2–12.
- Liu, D., Wang, X., Xu, Z., Li, Y., Liu, W., 2020. Space target extraction and detection for wide-field surveillance. *Astron. Comput.* 32, 100408. doi:10.1016/j.ascom.2020.100408.
- Mortari, D., Bruccoleri, C., Samaan, M., 2004. The pyramid star identification technique. *Navigation* 51, 171–183.
- Na, M., Zheng, D., Jia, P., 2009. Modified grid algorithm for noisy all-sky autonomous star identification. *IEEE Trans. Aerosp. Electron. Syst.* 45, 516–522.
- Nardino, V., Guzzi, D., Buresi, M., Cecchi, M., Cecchi, T., Corti, F., Corti, M., Franci, E., Guidotti, G., Pippi, L., et al., 2019. MINISTAR: a miniaturized device for the test of star trackers. In: *International Conference on Space Optics—ICSO 2018*. Vol. 11180, SPIE, pp. 2821–2827.
- Nuno, F., Jones-Wilson, L., Mohan, S., Lo, K., Jones-Wilson, W., 2017a. Miniaturized star tracker stimulator for closed-loop testing of CubeSats. *J. Guid. Control Dyn.* 40.
- Nuno, F., Jones-Wilson, L., Mohan, S., Lo, K., Jones-Wilson, W., 2017b. Miniaturized star tracker stimulator for closed-loop testing of cubesats. *J. Guid. Control Dyn.* 40 (12), 3239–3246.
- OpenCV, 2014. *OpenCV-Python tutorials*. https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html. Accessed: 2022-11-10.
- Padgett, C., Kreutz-Delgado, K., Udomkesmalee, S., 1997. Evaluation of star identification techniques. *J. Guid. Control Dyn.* 20, 259–267.
- Patat, F., Moehler, S., O'Brien, K., Pompei, E., Bensby, T., Carraro, G., de Ugarte Postigo, A., Fox, A., Gavignaud, I., James, G., Korhonen, H., Ledoux, C., Randall, S., Sana, H., Smoker, J., Stefl, S., Szeifert, T., 2011. Optical atmospheric extinction over cerro paranal. *Astron. Astrophys.* 257, 14.
- Pedrotty, S.M., 2022. COTS camera and computer-based open-source star tracker. In: 44th Annual AAS Guidance, Navigation and Control (GN&C) Conference.
- Pedrotty, S., Lovelace, R., Christian, J., Renshaw, D., Quintero, G., 2020. Design and performance of an open-source star tracker algorithm on commercial off-the-shelf cameras and computers. In: *Annual AAS Guidance, Navigation and Control Conference*. JSC-E-DAA-TN77360.
- Perryman, M.A.C., 1997. *The Hipparcos and Tycho Catalogues*, Volume 1. Tech. Rep., ESA.
- Riesing, K.M., Yoon, H., Cahoy, K.L., 2018. Rapid telescope pointing calibration: a quaternion-based solution using low-cost hardware. *J. Astron. Telesc. Instrum. Syst.* 4 (3), 034002. doi:10.1117/1.JATIS.4.3.034002.
- Rijlaarsdam, D., Yous, H., Byrne, J., Oddenino, D., Furano, G., Moloney, D., 2020. A survey of lost-in-space star identification algorithms since 2009. In: MDPI. p. 18. doi:10.3390/s20092579.
- Rufener, F., 1986. The evolution of atmospheric extinction at La Silla. *Astron. Astrophys.* 165, 12.
- Samaan, M., Steffes, S., Theil, S., 2011. Star tracker real-time hardware in the loop testing using optical star simulator.
- Schaefer, Bradley E., 1986. Atmospheric extinction effects on stellar alignments. *Archaeoastronomy* 10, 11.
- Schiattarella, V., Spiller, D., Curti, F., 2017. A novel star identification technique robust to high presence of false objects: The multi-poles algorithm. *Adv. Space Res.* 59, 2133–2147.
- Shuster, Malcolm D., 1978. Approximate algorithms for fast optimal attitude computation. In: *AIAA Guidance and Control Conference*.
- Shuster, Malcolm D., 1981. Three-axis attitude determination from vector observations. *J. Guid. Control* 4, 70–77.
- Silani, E., Lovera, M., 2006. Star identification algorithms: Novel approach & comparison study. *IEEE Trans. Aerosp. Electron. Syst.* 42, 14.
- Spratling, B.B., Mortari, D., 2009. A survey on star identification algorithms. In: MDPI. pp. 94–108. doi:10.3390/a2010093.
- Sreejith, A.G., Mathew, J., Sarpotdar, M., Mohan, R., Nayak, A., Safonova, M., Murthy, J., 2014. A raspberry Pi-based attitude sensor. *J. Astron. Instrum.* 03 (02), 1440006. doi:10.1142/S2251171714400066.
- Toloei, A., Zahednamazi, M., Ghasemi, R., Mohammadi, F., 2020. A comparative analysis of star identification algorithms. *Astrophys. Space Sci.* 365, 10. doi:10.1007/s10509-020-03775-9.
- Vyas, A., Roopashree, M.B., Prasad, B.R., 2009. Performance of centroiding algorithms at low light level conditions in adaptive optics. *IEEE Comput. Soc.* 366–370. doi:10.1109/ARTCom.2009.30.
- Wei, X., Wen, D., Song, Z., Xi, J., Zhang, W., Liu, G., Li, Z., 2019. A star identification algorithm based on radial and dynamic cyclic features of star pattern. *Adv. Space Res.* 63, 2245–2259. doi:10.1016/j.asr.2018.12.027.
- Wei, X., Zhang, G., Jiang, J., 2009. Star identification algorithm based on log-polar transform. *J. Aerosp. Comput. Inf. Commun.* 6, 483–490.



DSPACE

<https://dspace.org/>

Calibration and testing strategies to correct atmospheric effects on star tracking algorithms

Jannin, Louis; Felicetti, Leonard

2024-01-01

Elsevier

<https://doi.org/10.1016/j.ascom.2024.100794>

Jannin L, Felicetti L. (2024) Calibration and testing strategies to correct atmospheric effects on star tracking algorithms. Astronomy and Computing, Volume 46, January 2024, Article number 100794