



POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Object Detection and Image Cropping Integration on a Tracklet Extraction Pipeline

TESI DI LAUREA MAGISTRALE IN
SPACE ENGINEERING - INGEGNERIA SPAZIALE

Author: **Lorenzo Longo**

Student ID: 927958
Advisor: Prof. Pierluigi Di Lizia
Co-advisors: PhD Candidate Andrea De Vittori,
PhD Candidate Riccardo Cipollone
Academic Year: 2021-22

Abstract

Space Surveillance and Tracking activities are fundamental for maintaining an adequate level of safety in the space environment, allowing the actors in the field to keep their instruments operational. Given the buildup of high-speed clutter and the ever-increasing number of inoperative satellites, present and future space missions are at stake due to recurring conjunction events between resident space objects and strategic orbiting assets. The main activities of the Space Surveillance and Tracking segment are the observation of objects in orbit and the maintenance of an updated catalog with their orbital parameters. This allows us to maintain collision risk awareness and be able to perform orbit modification maneuvers to avoid potential impacts.

Monitoring space objects is mainly performed with sensor networks of combined radar and optical ground stations. The latter relies on advanced image processing techniques to locate the streak an orbiting object illuminated by the Sun leaves on the detector during observation campaigns.

In this thesis, an alternative structure of an existing pipeline for tracklet extraction is proposed. In particular, the core of this work is to integrate a Machine Learning aided object detection suite for tracklet spotting with a tool for astrometric reduction. Via object detection, it is possible to crop a night shot around the object of interest and perform the tracklet extraction to obtain the centroid position. By carrying out the astrometric reduction of the image, the angular coordinates expressed in Right Ascension and Declination are associated with the pixel coordinates of the centroid of the tracklet. This data can be used to initiate the orbital determination process.

The results obtained from this customized structure are compared with those obtained from astrometric reduction alone. It is observed a lowered computational time without losing the ability to correctly analyze a large number of images and to accurately estimate the centroid angular coordinates.

Keywords: Space Surveillance and Tracking, Space Debris, Object Detection, Tracklet Extraction, Astrometric Reduction

Abstract in lingua italiana

Le attività di sorveglianza e tracciamento sono fondamentali per il mantenimento di un'adeguato livello di sicurezza nell'ambiente spaziale, consentendo agli attori in campo di mantenere operativi i propri strumenti. Data la crescente congestione di oggetti orbitanti e l'aumento del numero di satelliti non operativi, il successo delle missioni spaziali presenti e future è a rischio a causa di ricorrenti eventi di congiunzione tra oggetti spaziali residenti e risorse strategiche in orbita. Le attività principali del segmento di Space Surveillance and Tracking sono l'osservazione degli oggetti in orbita e il costante aggiornamento di un catalogo con i loro parametri orbitali. Questo consente di avere consapevolezza del rischio di collisione e di essere in grado di eseguire manovre di modifica dell'orbita per evitare potenziali impatti.

Il monitoraggio degli oggetti spaziali avviene principalmente integrando reti di sensori che includono stazioni terrestri radar e ottiche. Queste ultime si basano su tecniche avanzate di elaborazione delle immagini al fine di individuare la traccia che un oggetto orbitante illuminato dal Sole lascia sul sensore durante le campagne di osservazione.

In questa tesi viene proposta una struttura alternativa di una esistente pipeline per l'estrazione delle tracce. In particolare, il fulcro di questo lavoro è l'integrazione di uno strumento per il rilevamento di oggetti basato su tecniche di Apprendimento Automatico che permette di rilevare la traccia, con uno strumento per la riduzione astrometrica. Attraverso lo strumento di rilevazione, è possibile ritagliare uno scatto notturno attorno all'oggetto di interesse ed eseguire l'estrazione della traccia per ottenere la posizione del centroide. Effettuando la riduzione astrometrica dell'immagine, le coordinate angolari espresse in Ascensione Retta e Declinazione sono associate alle coordinate pixel del centroide della traccia. Questi dati possono essere utilizzati per avviare il processo di determinazione orbitale. I risultati ottenuti da questa pipeline modificata vengono confrontati con quelli ottenuti dalla sola riduzione astrometrica. Si osserva un tempo computazionale ridotto senza perdere la capacità di analizzare correttamente un elevato numero di immagini e di stimare accuratamente le coordinate angolari del centroide.

Parole chiave: Sorveglianza Spaziale e Tracciamento, Detriti Spaziali, Rilevamento di

Oggetti, Segmentazione, Riduzione Astrometrica

Contents

Abstract	i
Abstract in lingua italiana	iii
Contents	v
1 Introduction	1
1.1 Orbital overpopulation problem	1
1.2 Space Surveillance and Tracking	3
1.3 State of the art	4
1.4 Thesis purpose and workflow	7
2 Fundamentals	9
2.1 Reference frames	9
2.2 Optical telescopes	12
2.3 Tracklet detection with RID	17
2.4 Astrometric calibration with Astrometry.net	21
2.5 Tracklet extraction with AstroDART	22
3 Tracklet extraction introducing object detection and image crop	27
3.1 Pipeline architecture	27
3.2 Analysis of images converted from FITS to PNG format	32
3.3 Analysis of images in FITS format	34
4 Results	37
4.1 Pipeline performances analyzing images in PNG format	37
4.2 Pipeline performances analyzing images in FITS format	51
5 Conclusions and future developments	61

Bibliography	63
A Appendix A	67
B Appendix B	71
List of Figures	87
List of Tables	91
List of Symbols	93
List of Acronyms and Abbreviations	95
Acknowledgements	97

1 | Introduction

This first chapter provides an introduction to the problem of orbital overpopulation due to the increasing presence of space debris in orbit, and how the Space Surveillance and Tracking (SST) segment deals with it as part of the European Space Agency (ESA) Space Situational Awareness (SSA) programme. Then, the focus shifts to the state-of-the-art techniques for tracklets identification, outlining strengths and weaknesses of the approaches presented in the literature. Ultimately, the thesis goal and organization is detailed.

1.1. Orbital overpopulation problem

Since the beginning of the "space age", started with the launch of Sputnik-1 in 1957, the number of artificial objects around the Earth has progressively grown. The most recent data provided by ESA counts more than 6200 launches and more than 8800 satellites in orbit, about 6200 of which are active [1]. The most concerning problem, however, is the presence of a large number of other non-functional mission-related objects in the form of spent rocket stages, launch adapters, lens covers, stray bolts and paint chips, solid rocket motor slag, etc., together with other smaller scattered fragments from collisions and explosions of spacecraft and upper stages. All these objects are defined as space debris. Some of the artificial objects in orbit (about 31500) are regularly tracked by Space Surveillance Networks and maintained in their catalog, but many others are not. Figure 1.1 shows the evolution of the space environment in terms of the number of cataloged and asserted objects, i.e. not reported but known to exist in the space environment [2].

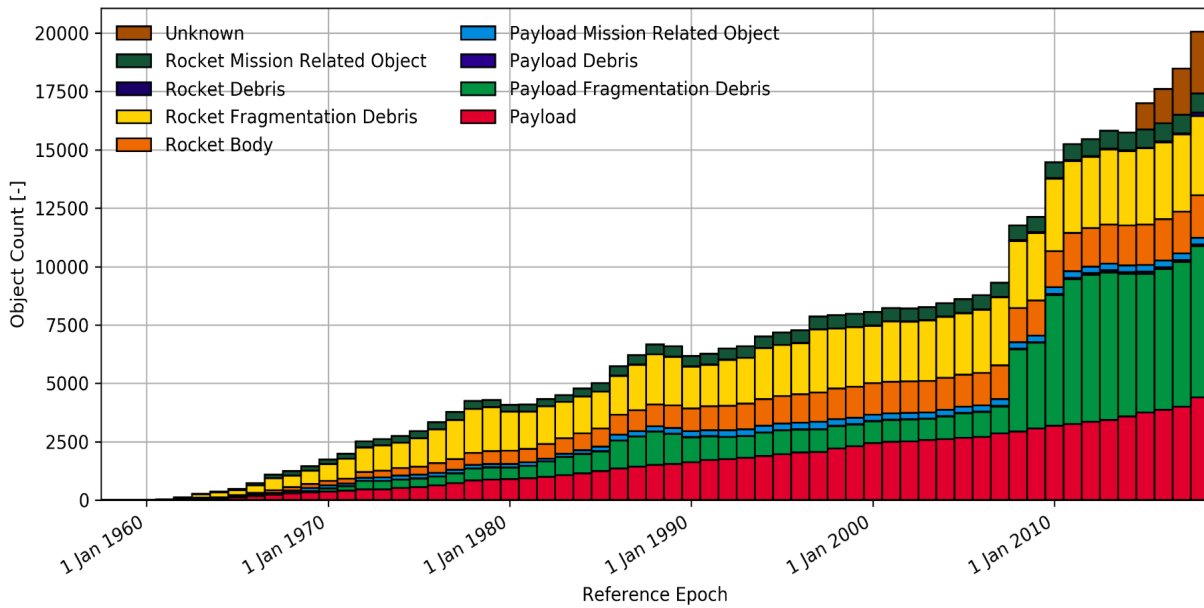


Figure 1.1: Evolution of space objects population [3]

From the previous graph, it is visible the harmful effect of a collision in orbit, such as the February 2009 run-in between a dead Russian Cosmos satellite and a commercial Iridium spacecraft, which produced an enormous amount of fragmentation debris.

The overall number of debris around the Earth is estimated based on statistical models to be [1]:

- 36500 space debris objects greater than 10 cm;
- 1000000 space debris objects from greater than 1 cm to 10 cm;
- 130 million space debris objects from greater than 1 mm to 1 cm;

They are mainly located in two orbital regimes [4]:

- Low Earth Orbit (LEO), including satellites orbiting the Earth at altitudes between 160 and 2000 kilometers. Fifty-five percent of all operational satellites are in this region since it is ideally situated for remote sensing missions, including Earth observation and reconnaissance, military objectives, and meteorology;
- Geosynchronous Equatorial Orbit (GEO), including satellites orbiting at an altitude of 35786 km, where the orbital period is equal to the Earth's rotational period. Around thirty-five percent of all the satellites belong to this region, making it the second most populated orbital regime. They are mainly exploited for telecommunications, defense, and meteorology.

These orbital regimes, and in particular the LEO one, are on the verge of becoming too crowded for comfort and the problem is now ready to get worse because of the rise of satellite “mega-constellations” requiring thousands of spacecraft, such as SpaceX’s Starlink [5].

Overpopulation of the common orbits increases the probability of collisions between satellites which would pollute the space environment with a high number of small-sized fragmentation debris. This cascading effect whereby the generation of space debris via collisions and explosions in orbit could lead to an exponential increase in the number of artificial objects in space was first postulated by Donald Kessler in 1978 [6] and takes the name of Kessler syndrome [2].

Orbit overpopulation represents then both a short-term risk of collision, faced by satellites present in a given orbital regime, as well as a long-term risk of triggering the Kessler syndrome.

1.2. Space Surveillance and Tracking

Under SSA programme, Space Surveillance and Tracking segment is concerned with watching for active and inactive satellites, discarded launch stages, and fragmentation debris orbiting the Earth. The core activity of the SST segment addresses a data catalogue containing information on all objects that have been detected in orbit. Processed data regularly updates the catalogue for collision probability assessment among the thousands of tracked objects and operational satellites. When necessary, issue warnings are sent to satellite controllers. Another SST application addresses the detection and decay prediction of large pieces of space debris, typically non-functioning satellites or upper stages, that may re-enter the atmosphere and possibly endanger people or infrastructure on the ground [7].

This requires a system of sensors comprising, typically, radars, telescopes and laser-ranging stations, and data centers to process the acquired observation data.

Any SST system can then be considered as a ‘processing pipeline’ to process observation data acquired by sensors and provide derived applications and services, typically comprising collision warnings [8].

A possible flow chart representative of the SST activities is shown in Fig. 1.2.

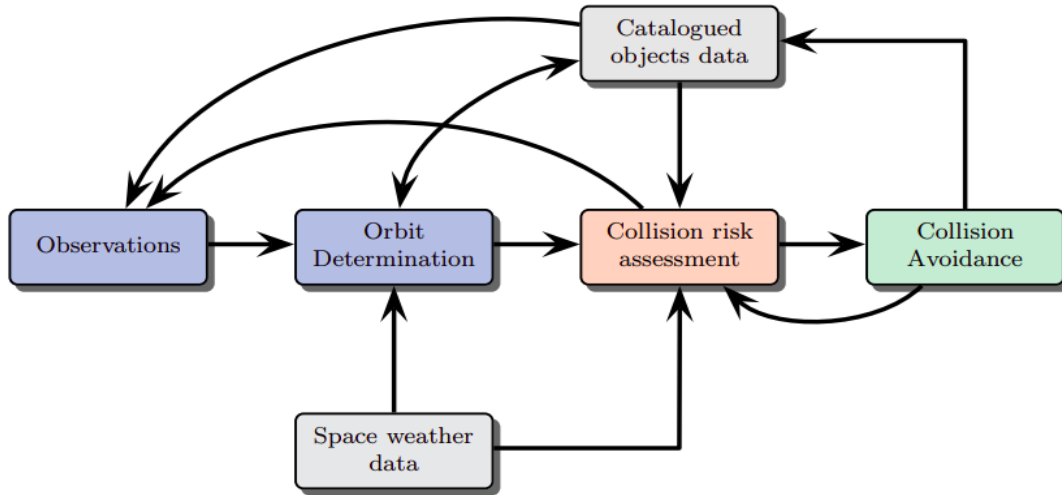


Figure 1.2: Space Surveillance and Tracking activities [9]

By looking at the previous graph it is clear that both the observation phase and the tracklet extraction phase, which is part of the orbit determination block, play a crucial role for SST activities, being the starting point for achieving high-level tasks such as collision risk assessment and collision avoidance.

1.3. State of the art

In a modern era, ground-based optical telescopes are one of the major observation methods to survey space debris, being it effective and feasible [10]. The observation of an object of interest with an optical telescope can be performed only if the following conditions are met [11]:

- the object must be above the station's horizon. This means that the elevation angle defined in the context of the Topocentric Horizon frame must be positive (see Sect. 2.1);
- the object must be illuminated by the Sun or an artificial light source;
- the object's brightness must exceed that of the background sky by a certain margin.

Optical telescopes can perform observations of space debris and satellites with two different approaches:

- staring (or survey) mode; the telescope is pointed towards the sky, moving at sidereal rate. Stars appear as fixed points on the background, while the debris or the satellite streaks to the field, resulting in a bright line on the image. The longer is the exposure

time, the longer is the resulting tracklet. An example is shown in Fig. 1.3a;

- chasing (or tasking) mode; once the sensor has detected an object, it starts moving at the same velocity of the target, acquiring the image. The object appears as a single dot, while stars streak through the field. An example is shown in Fig 1.3b.

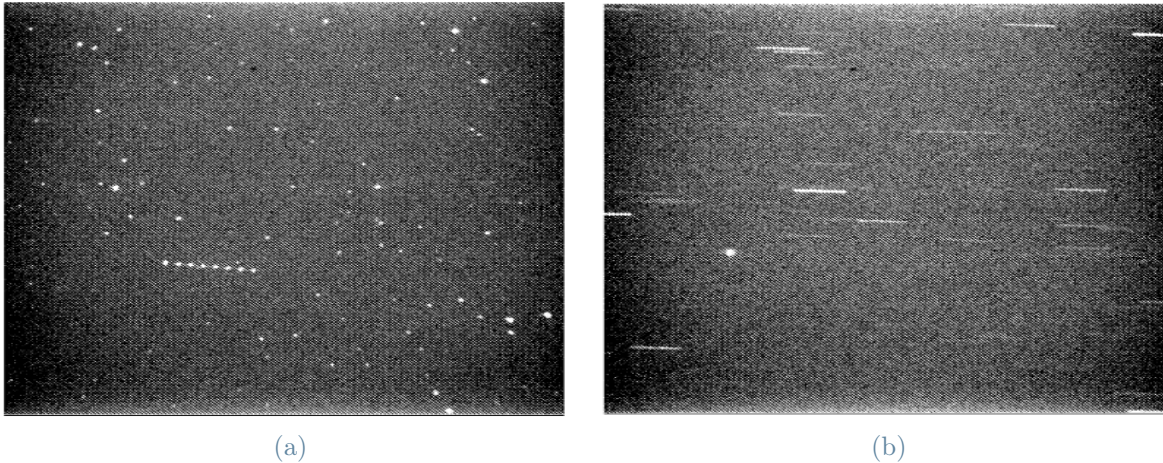


Figure 1.3: Staring (a) and chasing (b) acquisition modes [12]

Once the observation with an optical telescope is performed, some effort is devoted to the processing of trailed sources. Low astrometric accuracy at this stage is indeed the main cause of the orbital data uncertainty for space debris [10]. Normally, raw data cannot be immediately used for detection. Commonly, a pre-processing step devoted to photometric corrections is applied to enhance features of interest inside the image [13].

Some methods for the extraction of the tracklet have been developed such as thresholding, edge detection, some techniques based on Point-Spread-Function (PSF) fitting, and template matching.

Thresholding

Thresholding is a type of image segmentation, where the original image is converted to a binary mask image that splits regions (connected pixels) that are considered objects from those that are part of the background. The downside is that an educated guess must be made about an appropriate threshold value and this is not easy due because of noise, background variations, or diffuse edges of the objects; any chosen threshold may result in some true objects being overlooked (false negatives) and some spurious objects being considered as real (false positives). Thresholding can be used to create masks that select only the interesting parts of an image, or as the first step before edge detection.

Edge detection

Edge detection is an image processing technique for finding boundaries of objects within images. It works by detecting discontinuities in brightness and is used for image segmentation and data extraction in areas such as image processing and computer vision.

One of the most famous algorithms for edge detection is Automated Streak Detection for Astronomical Images (ASTRiDE) [14]. It is a Python package capable of detecting streaks in astronomical images with “boundary-tracing” technique. Any streak-like trace caused by the passage of moving objects like satellite, space debris or Near-Earth Objects (NEOs), can be detected with this algorithm, capable of quantifying the shape of each border to determine whether or not it is a streak. The usual steps are:

1. Background removal: ASTRiDE calculates background level and its standard deviation using sigma-clipped statistics; then, it removes the background from the original image.
2. Contour map: ASTRiDE derives the contour map of the original image. The level of the contour depends on the background standard deviation previously computed, through a user-defined multiplicative factor. The borders are then detected using the contour map.
3. Streak determination: the algorithm recognises and removes stellar sources using morphological parameters derived from each boundary and keeps the tracklets.

This kind of approach is more accurate, but it tends to fail at low Signal to Noise Ratio (SNR) also due to brightness variations along the tracklet [15].

Point Spread Function

When working with real images containing trailed sources, one has to deal with trails distorted by atmospheric turbulence. The PSF is the spatial distribution of the intensity of the source, and these distributions can be modeled by several different functions [16]. In this case, the basic idea of the PSF fitting procedure is to use a model profile of a trail extended along the direction inferred by either the apparent diurnal motion (for stars) or the apparent velocity of the object. This seems to be the most robust and versatile way to obtain the accurate trail positions [15]. Nevertheless, some specific issues may make PSF fitting fail or its accuracy decrease, such as overlapping or under-sampled trails, trails with low-SNRs, and trails surrounded by a rapidly changing background. Furthermore, PSF fitting method is more time consuming, which limits its application in real-time image processing [10].

Template Matching

Template Matching is a technique for object localization, which allows to identify parts of an image that match, under some criteria of similarity, an arbitrarily chosen image template. A template is a typical model or representative instance that is searched in an image; for space objects, a template should be a rectangular region containing a typical trail. In this method, a part of the actual observation image containing a bright trail is cut as a template; then, parts of the actual image that match the predefined template are identified. Normalized cross-correlation can be taken as the similarity measurement. Template Matching involves two critical steps, the similarity measurement and the search strategy. The drawback of template matching is its high computational cost which has two distinct origins. The first source of complexity is the necessity of using multiple templates to accommodate the variability exhibited by the appearance of complex objects. The second relates to the representation of the templates: the higher the resolution, i.e. the number of pixels, the more the computational burden [17]. When it comes to track objects in LEO, telescopes with a Field of View (FoV) of the order of 2-3° are commonly employed. Combining the relative small FoV and the high angular rate of the observed objects, a faster even if less accurate model would be beneficial for quick tracklet reconstruction and follow up observations scheduling [13].

1.4. Thesis purpose and workflow

The purpose of this thesis is to present and analyze a pipeline embedding a first Machine Learning (ML) based routine intended to locate the streak in FoV with a later centroid tracklet extraction refinement via edge detection techniques. Obtaining information about the position of the tracklet with the tool based on ML, it allows cropping the image and then limiting the tracklet extraction process to the portion of the image of interest. The performance is assessed by varying the dimensions of the cropped image to evaluate success rate of the tracklet extraction process, computational cost, and accuracy in determining celestial coordinates of the object.

The thesis is structured as follows: Chapter 2 deals with fundamental notions of space reference frames, optical telescopes, giving then valuable insights of the mentioned separate modules. In Chap. 3, the integration of the tools is presented and the performed analysis is described, considering different possible architectures of the overall pipeline. Moving on to Chap. 4, the results of the analysis are presented, considering different parameters of merit. Finally, in Chap. 5 some considerations based on the obtained results and possible future developments are discussed.

2 | Fundamentals

This chapter describes the fundamental concepts and tools necessary for the development of the thesis. Section 2.1 describes the main reference systems adopted in the astronomical and space fields. Section 2.2 illustrates the physical principle of optical telescopes, the image acquisition mode, and the technical specifications of the particular instrument from which the images analyzed in this thesis are obtained. It also describes the Flexible Image Transport System (FITS) format in which these images are saved before being analyzed. Section 2.3 reports a tool for detecting and locating tracklets within astronomical images, called Real Images Detector (RID), based on Machine Learning techniques. In Sect. 2.4, the functioning of Astrometry.net is described. This system, upon receipt of an astronomical image, returns its aiming, scale, and orientation. Lastly, Sect. 2.5 describes a tool named Astronomical Data Analysis and Recovery from Tracklets (AstroDART), designed for the detection and refinement of tracklets and the determination of the celestial coordinates of the object of interest at the time of observation.

2.1. Reference frames

To determine the position of an object, it is necessary to define a reference system and a coordinate system. A reference system implies an origin and three orthogonal axes of unitary length, respectively represented in Fig. 2.1 by the point O and the triad $\{\mathbf{i}, \mathbf{j}, \mathbf{k}\}$. A reference system is inertial if it does not undergo any acceleration.

A coordinate system is necessary to locate a point of interest within a reference system [18]. The most used ones for astronomy applications are:

- Cartesian, in which three spatial coordinates define the position of a point. Referring to Fig. 2.1a, the position of point P is defined as $OP = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$.
- Spherical, in which one spatial coordinate (r) and two possible angular coordinates, the polar angle (ϕ) measured from the zenith and the azimuth angle (θ) spanning in the plane orthogonal to the zenith direction, are enough to locate a point. Referring to Fig. 2.1b, the position of point P is $OP = r \sin \phi \cos \theta \mathbf{i} + r \sin \phi \sin \theta \mathbf{j} + r \cos \phi \mathbf{k}$.

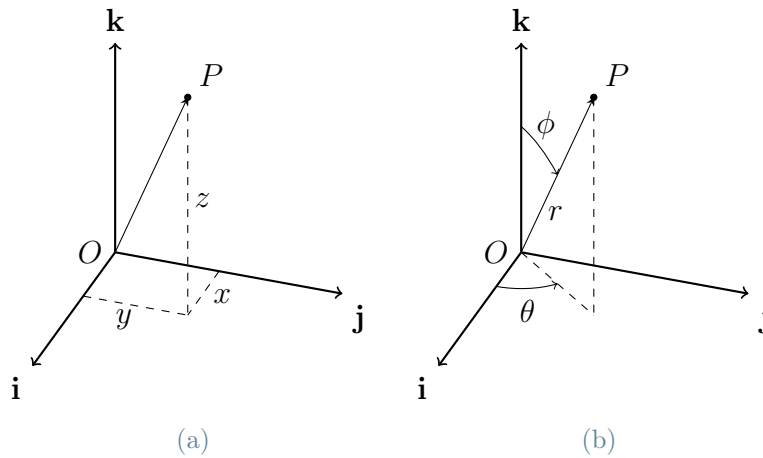


Figure 2.1: Cartesian and spherical coordinates [13]

Some of the most commonly employed reference systems for astronomy and space applications are the Earth Centered Inertial frame (ECI), the Topocentric Equatorial frame and the Topocentric Horizon frame. It is possible to switch from one reference system to another using appropriate rotation matrices.

Earth Centered Inertial

The ECI is a right-handed inertial¹ reference frame, with origin in the Earth's center of mass. Regarding Fig. 2.2, the X -axis points towards the vernal equinox direction at the J2000 epoch, the Z -axis coincides with the Earth's rotation axis and points northward and Y -axis completes the set. XY is Earth's equatorial plane. The green angle between the X -axis and the position vector's projection on the equatorial plane, measured counterclockwise, is the Right Ascension α , with values between 0° and 360° . The red angle from the equatorial plane to the object's location is the Declination δ , with values between -90° and $+90^\circ$.

Topocentric Equatorial Frame

A Topocentric frame is centered at the observer's location on the Earth's surface, rotating at its same angular velocity. Regarding Fig. 2.3, the Topocentric Equatorial frame has the three axes x, y, z coinciding with the X, Y, Z axes of the ECI frame; for this reason, both Right Ascension and Declination of the object of interest are the same as in Fig. 2.2.

¹acceleration towards the Sun is neglected.

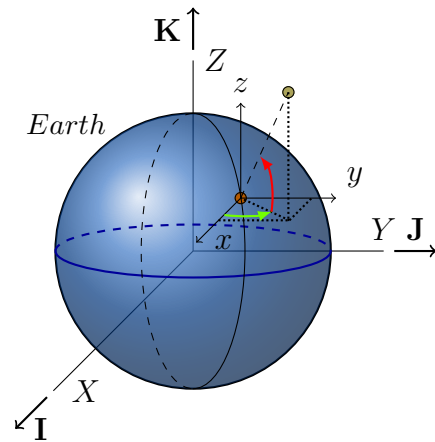
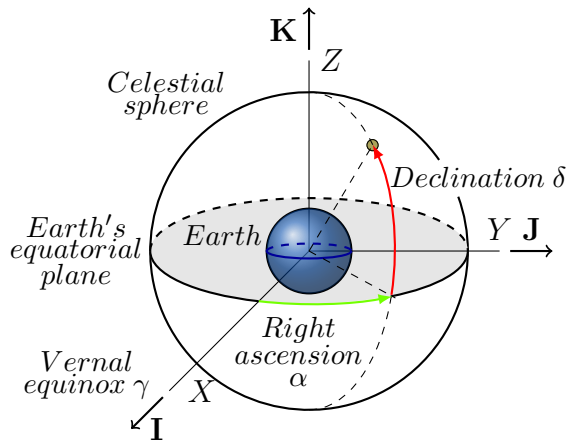


Figure 2.2: Earth Centered Inertial frame Figure 2.3: Topocentric Equatorial Frame

Topocentric Horizon Frame

The Topocentric Horizon frame is also centered at the observer’s location and its fundamental plane is the local horizon. The z -axis is normal to this plane and directed towards the zenith. A possible configuration is shown in Fig. 2.4a, with the x -axis pointing towards North and the y -axis pointing towards West; this frame is also named North West Zenith (NWZ). Concerning Fig. 2.4b, the two angular coordinates are the azimuth (β) and the elevation (ϵ). The former is measured clockwise from the North axis to the projection of the object’s position vector onto the local horizon plane; it ranges from 0° to 360° . The latter is measured from the local horizon plane to the object’s position vector; it ranges from -90° to $+90^\circ$. They play the same role of Right Ascension and Declination in the ECI reference frame.

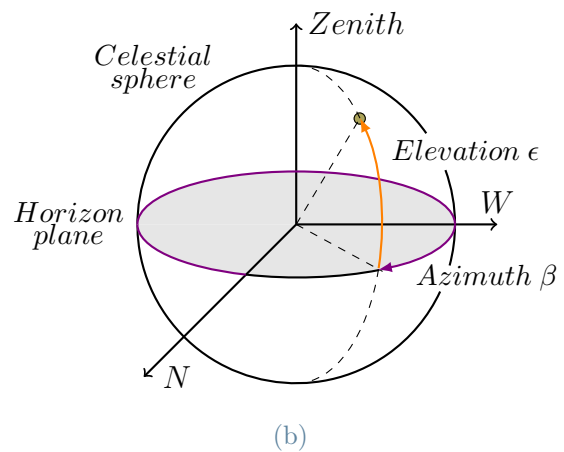
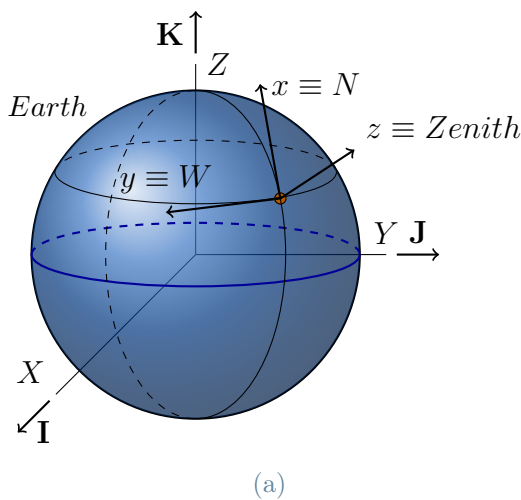


Figure 2.4: Topocentric Horizon Frame

2.2. Optical telescopes

Ground-based monitoring of satellites and space debris is performed thanks to a sensor network including both optical and radar instruments. This thesis revolves around the analysis of images obtained through telescope acquisitions.

An optical telescope fulfills three major tasks in astronomical observations [19]:

- it collects light from a large area, making it possible to study very faint sources;
- it increases the apparent angular diameter of the object, improving resolution;
- it is used to measure the positions of objects.

Working principle

A scheme with the working principle of an optical sensor is shown in Fig. 2.5.

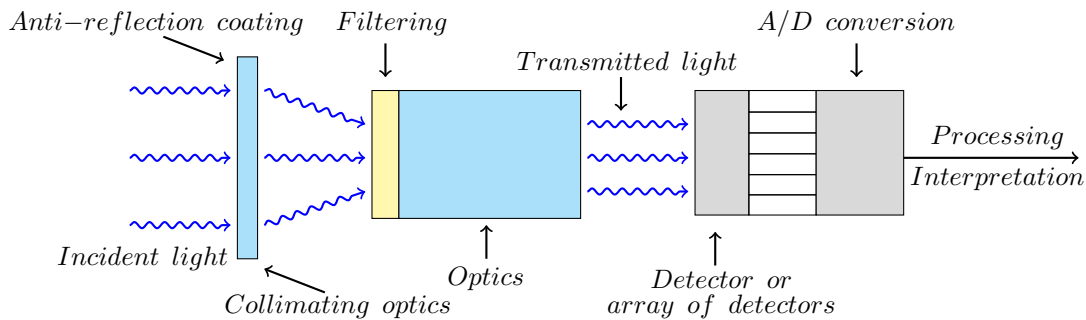


Figure 2.5: Optical sensor scheme [13]

An anti-reflection coating is used to ensure that all the light of interest goes into the sensor. The light is collected using static optics and collimated to fit the subsequent measurement acquisition subsystem. Further filters and collimating optics select subsets of wavelengths, known as spectral bands, which pass through and focus onto a detector that transforms photons to voltage. This voltage is amplified, digitized, and then processed. Charge Coupled Device (CCD) and Complementary Metal Oxide Semiconductor (CMOS), shown in Fig. 2.6, are the most widely used technologies for the realization of detectors. In both cases an electrical signal is generated from light thanks to the photoelectric effect; each sensing element (pixel) is filled with electrons when hit by photons until it stops being exposed to light. In a CCD sensor each pixel's charge is transferred through a limited number of output nodes to be converted to voltage, affecting the image output uniformity and quality. [20]. The drawback for a CCD sensor lies in the reduced reading speed, which is prominent for high resolution shots. [21]. Furthermore, a mechanical shutter is needed

for stopping receiving photons during the read-out. On the other hand, with the CMOS architecture, acquisition is almost instantaneous since each pixel has its charge-to-voltage conversion, granting a faster read out at the expense of uniformity.

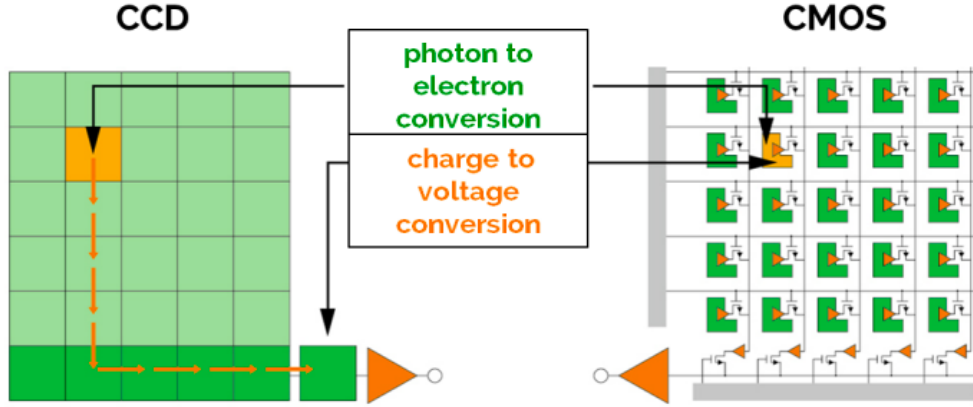


Figure 2.6: CCD and CMOS architectures [22]

A simplified model of a telescope, including the most relevant parameters, is shown in Fig. 2.7. The diameter of the objective is called aperture of the telescope (D). The focal length (f) is the distance between the telescope’s main optics (aperture) and the point where the image is created (focal point C). The ratio of the aperture to the focal length is called aperture ratio (D/f). This quantity characterizes the light-gathering power of the telescope. If the aperture ratio is large, near unity, one has a powerful, “fast” telescope and photographs can be taken using short exposures since the image is bright. A small aperture ratio (focal length much greater than aperture) means a “slow” telescope. The focal ratio is given by the focal length divided by the telescope’s aperture (f/D); it is the inverse of the aperture ratio and it is usually written as $f/number$. The Field of View (FoV) of a telescope instead is defined as:

$$FoV = 2 \arctan \frac{s}{2f} \quad (2.1)$$

where s represents the size of the detector.

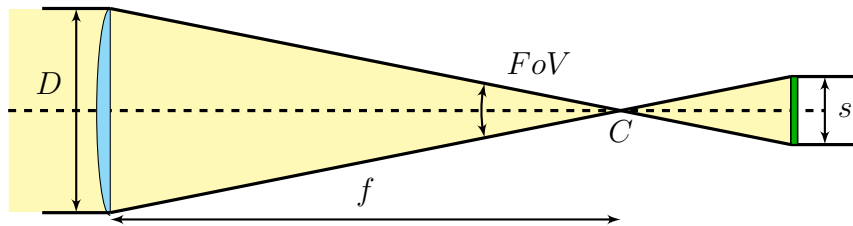


Figure 2.7: Simplified telescope model [11]

The light-collecting surface in a telescope is either a lens or a mirror and then optical telescopes are divided into two types, lens telescopes or refractors and mirror telescopes or reflectors [19]. These two configurations are shown in Fig. 2.8.

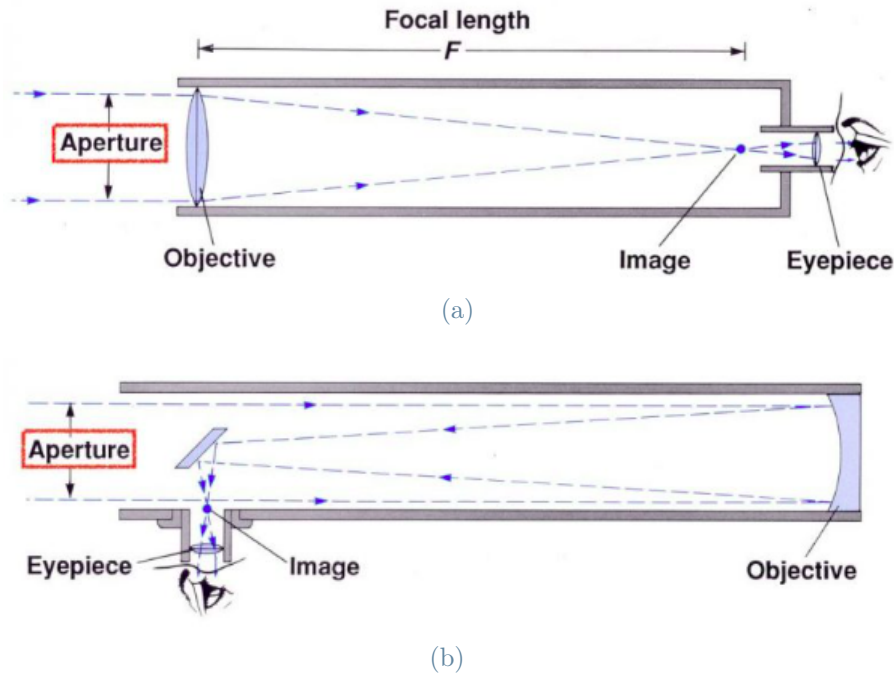


Figure 2.8: Refractor (a) and reflector (b) configuration [23]

Refractor telescopes, shown in Fig. 2.8a, have two lenses, the objective, which collects the incoming light and forms an image in the focal plane, and the eyepiece, a small magnifying glass for looking at the image. The main problem associated with a single refracting lens is chromatic aberration because glass refracts colors by different amounts. It can be corrected by combining three or even more lenses of different glasses in the objective.

The most common telescope type is the reflector, shown in Fig. 2.8b. As a light-collecting surface, it employs a mirror coated with a thin layer of aluminum. A parabolic mirror reflects all light rays entering the telescope parallel to the main axis into the same focal point. One of the advantages of reflectors is the absence of chromatic aberration [19]. Coma aberration is instead associated with this architecture, causing point sources appearing as comet-like radial smudges. Other limitations are the focus error due to the field curvature, astigmatism, an azimuthal variation of the focus around the aperture causing point source images off-axis to appear elliptical, and distortion of object shapes [21].

Ground-based telescopes

The great majority of astronomical observations are carried out from the surface of the Earth. With a ground-based telescope, the atmosphere affects observations in many ways. The most important effect is refraction, which changes the apparent altitudes of objects. Empirical formulas are used for taking into account and correcting it. Furthermore, the air is never quite steady and there are layers with different temperatures and densities; this causes convection and turbulence. When the light from a star passes through the unsteady air, rapid changes in refraction in different directions result. Thus, the amount of light reaching a detector constantly varies; the star is said to scintillate. A telescope collects light over a larger area to diminish scintillation. Instead, differences in refraction along different paths of light through the atmosphere smear the image, and point sources are seen in telescopes as vibrating speckles. This phenomenon is called seeing, and the size of the seeing disc may vary from less than an arc second to several tens of arc seconds. In astronomy one often has to observe very faint objects. Thus, the background sky must be as dark as possible, and the atmosphere as transparent as possible [19]. To reduce light pollution and minimize water vapor absorption, ground-based telescopes are located far from urban areas, at high altitudes, and in dry regions.

Pratica di Mare ground station

Italian Air Force (ITAF) has recently installed an optical sensor for SST in Pratica di Mare Air Force Base, named Pratica di Mare Military Telescope (PdM-MiTe); it is designed by GMSpazio and Officina Stellare and is able to observe the portion of space above it with coverage of $360^\circ \times 90^\circ$ in azimuth and elevation (see Fig. 2.9). It is built with an exclusive Riccardi-Honders flat field optical design, with an aperture of 350 mm and a focal ratio $f/2.8$ [24].



Figure 2.9: Pratica di Mare Military Telescope for SST [25]

It is equipped with two CCD sensors: one with a wide FoV used for surveillance tasks and the second with a narrow FoV of about 2° dedicated for tracking specific objects. The latter (from which the analyzed images are acquired) is an array of 4096 by 4096 pixels in resolution, leading to a 16 *Mpxl* detector. It measures nearly 37 *mm* on both sides. The telescope is mounted on a platform in equatorial layout designed for high-speed tracking and pointing purposes. The time sync is performed by a GPS sensor, which allows knowing the exact time of the shot. Once the image is acquired, a FITS file containing the main info of the passage is generated and then the image can be analyzed and processed for the extraction of orbital parameters.

Flexible Image Transport System format

Flexible Image Transport System (FITS) is a file format designed to store, transmit, and manipulate scientific images and associated data [26]. The term "image" in the standard's name is loosely applied and FITS files often contain only non-image data. Astronomers view their images as data for analysis rather than simply as pictures to look at. FITS format was designed to facilitate the unambiguous transmission of n-dimensional regularly spaced data arrays, an n-cube. These multi-dimensional arrays may be 1-D spectra, 2-D images, or data cubes of three or more dimensions. Two-dimensional tables containing rows and columns of data can also be stored in a FITS file. FITS is categorized primarily as a dataset format, with use for image data as secondary.

A FITS file consists of one or more Header + Data Units (HDUs), where the first HDU is called the primary HDU, or primary array. The primary array may be empty or contain an N-dimensional array of pixels, such as a 1-D spectrum, a 2-D image, or a 3-D data cube. The FITS files analyzed in this thesis are 2-D images with unsigned 16-bit integers

data type, therefore each pixel value ranges from 0 (absolute black) to 65535 (absolute white). The header of these files reports important pieces of information like:

- BITPIX: bit depth used to represent each pixel;
- NAXIS1, NAXIS2: number of pixels in the first and second axis;
- DATE-OBS: date and time expressed in Universal Time Coordinated (UTC) of the beginning of the observation;
- EXPTIME: exposure time in seconds;
- XPIXSZ, YPIXSZ: pixel width and height (after binning);
- JD: Julian Date at exposure start time;
- FOCALLEN: telescope focal length;
- APTDIA: telescope aperture diameter;
- OBJECT: name of the target (if present);
- OBJECTRA, OBJECTDEC: topocentric right ascension and declination of the center of the image;
- NOTES: additional information added by the user. In the analyzed images, if the observed body's NORAD ID² is known, it will be found here.

2.3. Tracklet detection with RID

In 2021, Calvi developed a ML-based tool for tracklet detection and localization, named Real Image Detector (RID) [27]. It was generated by training a Convolutional Neural Network (CNN) based on an object detection algorithm (YOLOv5) architecture. Its aim is to act as a filter to speed up the tracklet extraction processes: the observations resulting from image acquisitions can be initially analyzed by RID, and only the shots containing streaks can then be analyzed by a tracklet extraction system to reduce the total time.

A scheme with the main steps performed by RID is shown in Fig. 2.10.

²NORAD ID is a progressive 5-digit number assigned to all satellites in orbit around the Earth in order of identification.

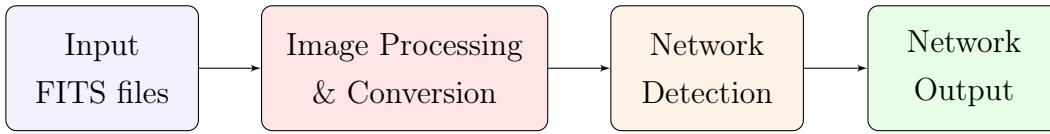


Figure 2.10: RID functioning scheme [27]

As a first step, images in FITS format acquired by the telescope are given as input to the tool, which then converts them into Portable Network Graphics (PNG) format to make them compatible with YOLOv5 architecture. The image processing step is necessary for highlighting the moving object and for reducing noise and vignetting in the original image; it is performed through the Scikit-Image Python library to obtain high quality images in the detection phase. This step is performed by the trained CNN and consists in identifying streaks within the image, when these are present. In case a tracklet is identified, a bounding box is generated around it. The network can distinguish tracklets from clouds, decreasing false positive detections. At the end of the pipeline, the tool outputs a copy of all images with bounding boxes printed around tracklets and clouds, and a folder storing the text files referred to analyzed images in which an object of interest is detected. These contain the parameters that allow locating a bounding box inside the image, such as its height, width and center's coordinates.

Figure 2.11 reports an example of what is obtained in output from the detection: specifically, Fig. 2.11a shows the processed image with the detected trail, while Fig. 2.11b depicts the geometrical parameters of the bounding box. The class of the detected object is reported in column 1 (1 for clouds and 0 for tracklets), the x and y coordinates of the bounding boxes' center are in column 2 and 3, and the width and height are in columns 4 and 5.

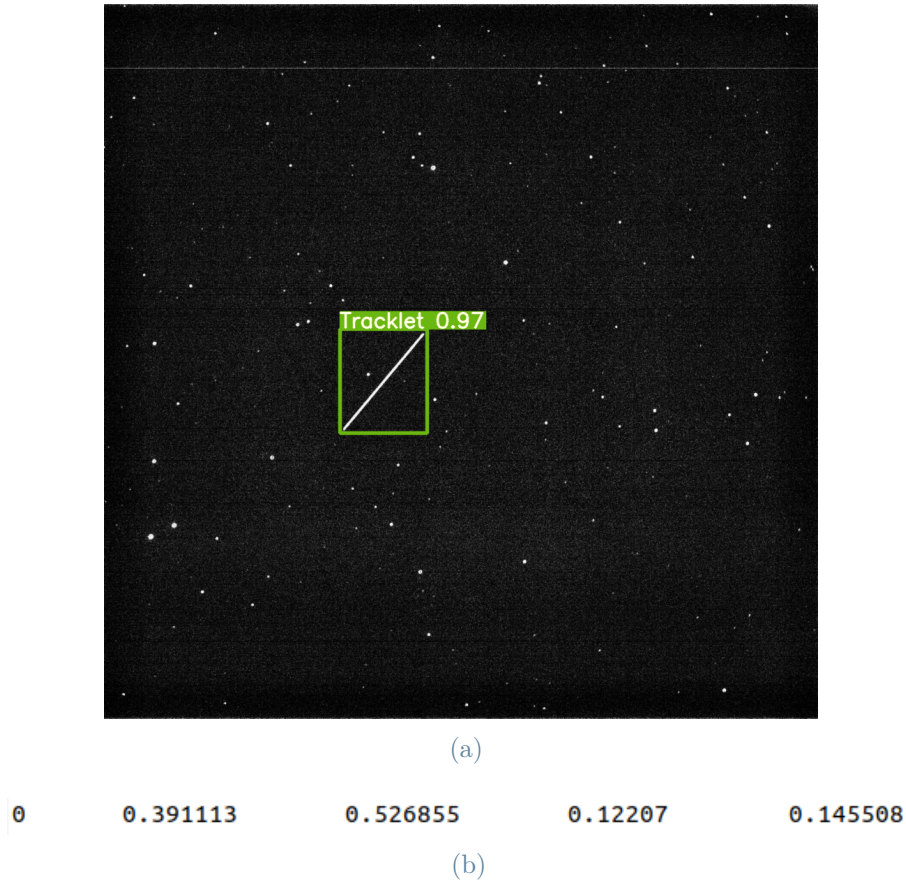


Figure 2.11: Tracklet detection output

Scikit-image processing

Image processing for high quality tracklet detection is implemented with Scikit-image (shortened skimage) Python library. The input FITS image is treated as a 2-D array, which undergoes a series of transformation:

- Logarithmic correction transforms the input image pixelwise according to the equation

$$O = g \cdot (2^I + 1) \quad (2.2)$$

where O and I are respectively the pixel values of the output and input image, while g is a constant;

- Contrast Limited Adaptive Histogram Equalization (CLAHE), an algorithm for local contrast enhancement that uses histograms computed over different tile regions of the image. Local details can then be enhanced even in regions that are darker or lighter than most of the image;

- Gamma Correction, also known as Power Law Transform. This function transforms the input image pixelwise according to the equation

$$O = I^\gamma \quad (2.3)$$

after scaling each pixel to the range 0 to 1. O and I are again the pixel values of the output and input image, while γ is a non negative real number;

- Down-conversion of the image to 8-bit unsigned integer format;
- Noise removal through median filtering technique. Each pixel value of the input image is replaced with the median of the neighboring pixel values;
- Image resize, that consists of scaling the image by a certain scale factor. In this case a factor 4 is used. Original images have an initial resolution of 4096 by 4096 pixels and are reduced to a resolution of 1024 by 1024 pixels.

Matplotlib.pyplot processing

A different image processing for fast even if slightly inaccurate detection is implemented with Matplotlib.pyplot library. In this case, the original FITS image is downscaled to a resolution of 512 by 512 pixels keeping the same color range of values, and an anti-aliasing filter is applied. Then, the following transformations are applied:

- definition of the data range that the colormap covers. It emphasizes low light sources but at the same time can generate noise;
- Lanczos interpolation, which consists of re-pixeling the image to make it more uniform and more grainy, to improve the quality of the tracklets. This interpolation method provides the best properties when it comes to preserving detail, and minimizing aliasing and artifacting. It increases the perceived sharpness of the image.

Figure 2.12 shows a comparison between the two above-mentioned image processing techniques, applied to the same image. In particular, Fig. 2.12a highlights the effect of skimage processing, while Fig. 2.12a the outcome of matplotlib processing. The former has more details, greater sharpness, and better resolution [27].

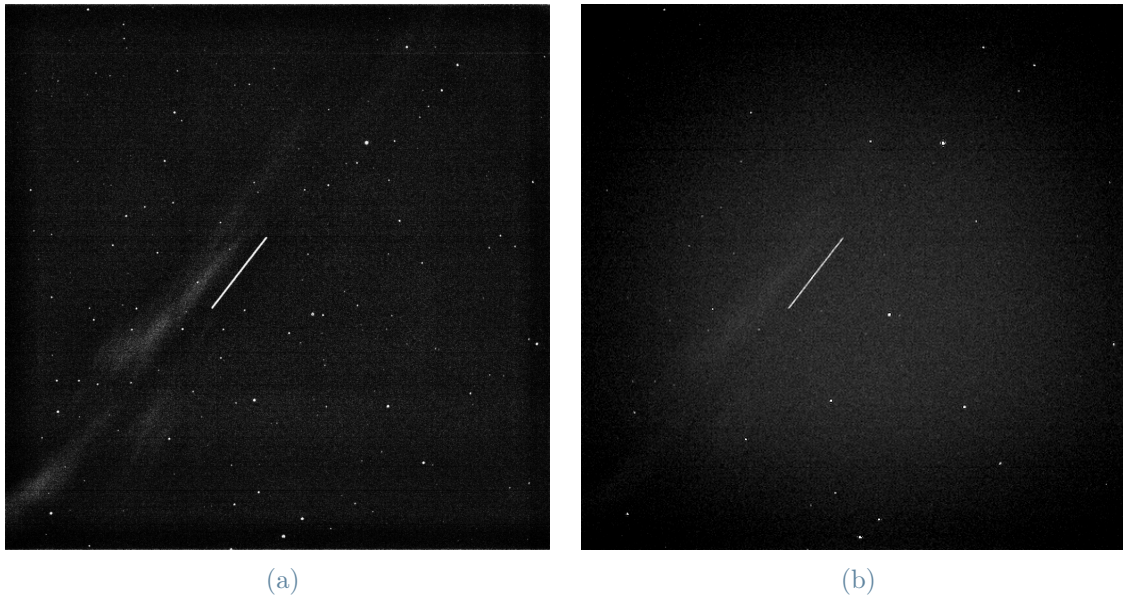


Figure 2.12: Effect of different image processing

2.4. Astrometric calibration with *Astrometry.net*

Astrometric calibration can be defined as a mathematical transformation relating the x, y positions of pixels in the image array to celestial coordinates in the sky, i.e. right ascension and declination. These transformations between sets of coordinates are standardized as World Coordinate System (WCS) conventions, defined in a set of formal papers, four of which have been officially approved as part of the FITS standard by the International Astronomical Union (IAU) [28]. A useful tool for performing this step is *Astrometry.net*. It is a system that takes an astronomical image as input and returns the pointing, scale and orientation of that image as output; it can reach a success rate slightly below 100%, with no false positive matches [29]. To achieve this, once a query image is given, the tool performs the following steps:

- astronomical sources (stars) detection, by performing some image processing;
- geometric hash code definition for each subset of these stars. The hash code describes the stars' relative positions. Subsets of four stars, named "quads", are used;
- research in pre-computed indexes for almost identical hash codes, which are expressed as a hypothesized match;
- verification criterion, phrased as a Bayesian decision problem, to decide if the hypothesized alignment is correct.

At the end of the astrometric calibration process, a new FITS file is produced, the header of which contains also the WCS data generated by *Astrometry.net*.

The example in Fig. 2.13 shows an image which is correctly solved by the astrometric calibration tool. Red circles enclose the detected astronomical sources, while green circles enclose stars listed in known indexes, the match of which is needed to solve the image.

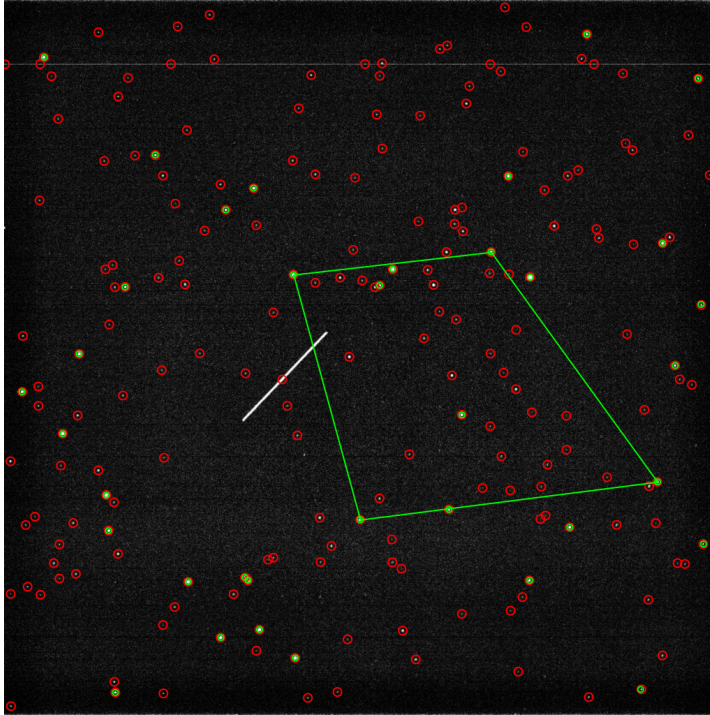


Figure 2.13: Astrometric calibration example

2.5. Tracklet extraction with AstroDART

AstroDART is a tool developed by González López-Cepero in 2022. After the astrometric reduction step performed with *Astrometry.net*, this tool detects the tracklet with ASTRiDE, then refines it, obtains the celestial coordinates of the object at the observation epoch, and performs telescope calibration by computing time bias after comparing the observations of known objects with catalog data [30]. If the telescope is pointing to an unknown object (no NORAD ID information in the NOTES key inside the FITS file header) the last step can not be performed. The tool is also designed to provide TDM files that contain the Right Ascension and Declination (RADEC) evolution of the object for future acquisitions, and to compute the light curve of the tracklet, but these two modules are beyond the scope of the analysis developed in this thesis so they have been deliberately excluded from the pipeline.

Figure 2.14 shows a chart that resumes AstroDART workflow, including only the functions of interest in this context.

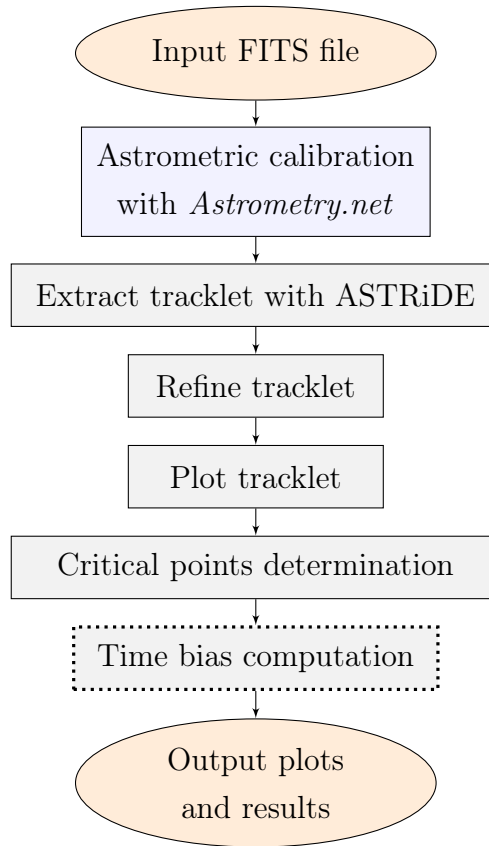


Figure 2.14: AstroDART working principle [30]

The tracklet detection process inside AstroDART is done via the Python package ASTRiDE, which uses boundary-tracing techniques to find all borders in an image, quantifying their shape to determine whether or not it is a streak-like trace left in the image by some fast-moving object like debris or a satellite [14]. ASTRiDE’s search for tracklets is ruled by some parameters that must be chosen by the user, the most relevant of which are:

- `contour_threshold`: is the threshold to extract a contour map. If this value is high, only bright streaks will be detected;
- `shape_cut`: defines an upper limit for the circularity of the detected contour. The circularity of streak-like objects is much smaller than that of a circle, which is 1;
- `radius_dev_cut`: defines a lower limit for the approximate deviation from roundness of the detected objects. A circle has a deviation from roundness of 0;
- `area_cut`: defines a lower limit for the area (i.e., the number of pixels) inside the

detected contour.

The tracklet search is carried out iteratively until just one is detected or the selected maximum number of iterations is reached. At each iteration, if many tracklets were detected at the previous step, the "contour_threshold" value is increased, while it is decreased if none was detected. If one of these two conditions happens at the final iteration, the analysis stops.

The following step is the refinement of the detected tracklet: this is an iterative process in which each long side of the tracklet is approximated to a straight line through a least-squares approach; then, the root mean square error of the contour points' distance with respect to this straight line is computed and those points whose distance is higher than a chosen threshold are deleted. This is done recursively until a maximum number of iterations is reached or no more points are deleted. This allows to obtain a tracklet with a more regular shape, decreasing the effect of the contamination in its contour due to the presence of foreign objects or image noise. At this point, both the raw and the refined tracklet can be plotted. An example is given in Fig. 2.15, where Figure 2.15a shows how a raw tracklet appears before the refinement process, and Figure 2.15b shows the same tracklet after the refinement step.

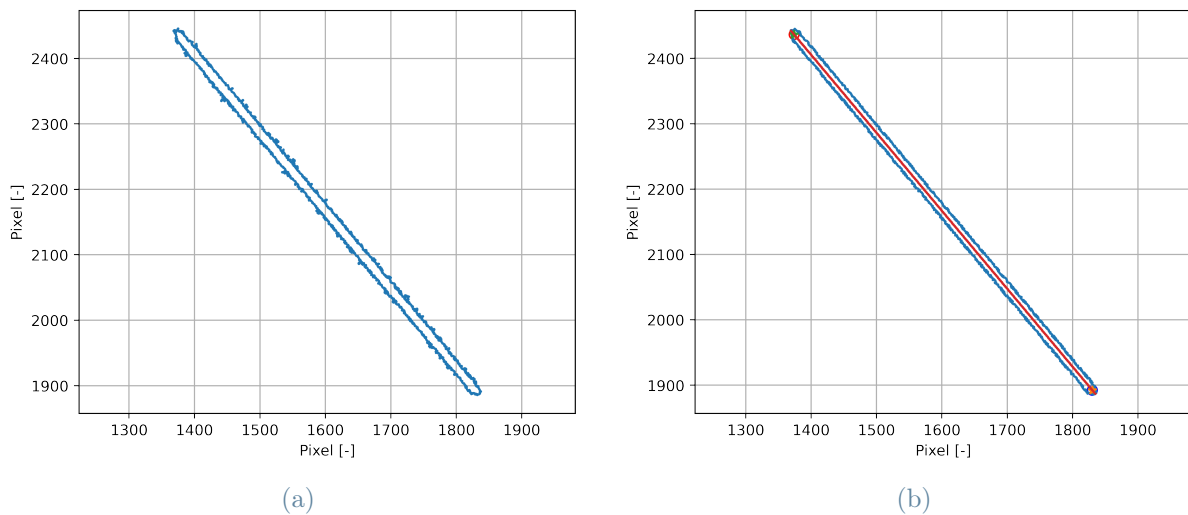


Figure 2.15: Tracklet aspect before (a) and after (b) the refinement process

The next step is the determination of the tracklet's initial point, centroid and last point. The most relevant between these three points is the centroid, whose position is obtained as the arithmetic mean of the coordinates of the refined tracklet's contour points. The successive time bias computation can indeed be performed by taking as reference both the centroid's position and the extremes points' position. In the work developed by González

López-Cepero [30] it is highlighted that the centroid-based analysis is the best approach to compute the time bias of the telescope since it gives more accurate results. Given the pixel coordinates of a fixed point of the tracklet, it is possible to obtain the RADEC coordinates by using the WCS data obtained from the astrometric calibration step.

After the critical points are determined, if the time bias calculation is possible (if the NORAD ID of the object is known), it is performed. The time bias is the delay between the timestamp of the image and the real acquisition epoch. The estimation of the time bias consists in a minimization process of the residuals between the RADEC coordinates of the chosen tracklet point (in this case the centroid) and the RADEC coordinates of the object at that epoch plus the delay, obtained from the catalog. The epoch associated with the centroid is the one corresponding to the observation timestamp plus half of the exposure time.

Finally, all output results are stored as files. For each analyzed image, in addition to the plot of the tracklet before and after the refinement, information on the RADEC coordinates and the residuals associated with the critical points are printed on a text file, both before and after time bias correction, assuming the latter is available.

3 | Tracklet extraction introducing object detection and image crop

This chapter presents the structure of an alternative pipeline that includes the upstream use of the RID tool to obtain the position of the bounding box that encloses the tracklet before the extraction process is performed with AstroDART. The organization of the pipeline and the analysis carried out on a set of telescope-acquired images are presented in Sect. 3.1.

Two different approaches are then considered. The first one consists in giving as input to the tracklet extraction tool the images that are previously converted from their original FITS format to the PNG format; this approach is discussed in Sect. 3.2. Concerning the second approach, once the step of object detection is completed with RID, the original images in FITS format are cropped around the tracklet, then the extraction process is carried out. This process is described in Sect. 3.3.

3.1. Pipeline architecture

The original pipeline developed by González López-Cepero in his Master's thesis [30], summarized in Fig. 2.14, takes as input the entire FITS format image that is obtained from a telescope's acquisition campaign. It should be noted that there is no prior knowledge on the presence of a space object streak in the input image. Not only does a successful iterative search depend on the actual presence of a target, but also on the ASTRiDE algorithm's effectiveness to detect it correctly.

An alternative pipeline is here proposed, with the upstream use of RID for obtaining information regarding both the possible presence of one (or more) tracklet(s) and its (their) position within the image. By knowing the position of the bounding box that encloses the tracklet, it is possible to make a crop of the image around it, so that the astrometric calibration and the tracklet extraction steps are performed only on the target portion of

the image.

A block diagram representing the architecture of the pipeline is shown in Fig. 3.1.

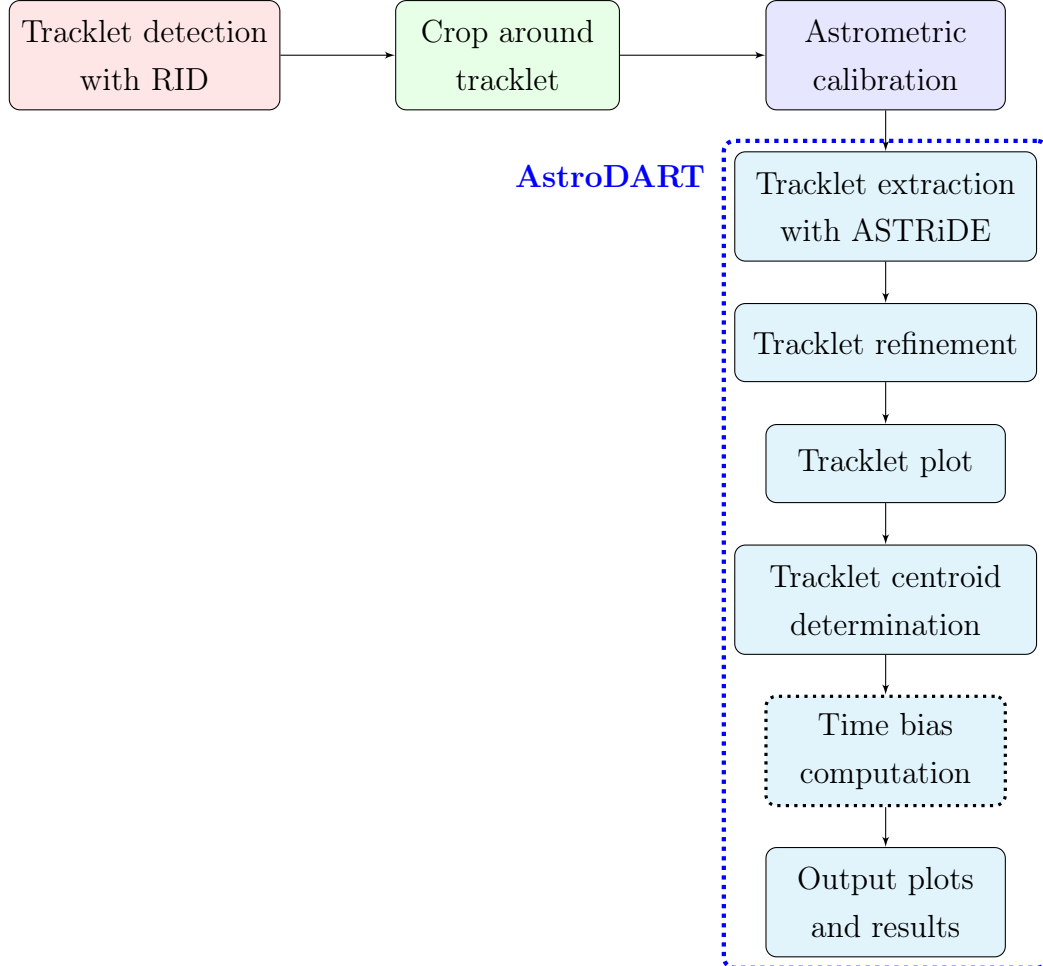


Figure 3.1: Pipeline integrating RID and AstroDART tools

The architecture shown in Fig. 3.1 should be advantageous for different reasons. First, the upstream introduction of the RID tool into the pipeline allows filtering images that do not contain any trail and therefore do not need to undergo the tracklet extraction process, saving a considerable amount of time. The iterative search performed with ASTRiDE after the astrometric calibration step is indeed quite onerous from a computational point of view, especially if no tracklet is present in the image and the search is repeated until the maximum number of iterations is reached.

Furthermore, the information provided as output by the RID tool regarding the position of the tracklet allows cropping the image, in such a way to include only a portion of the background sky; in this way, it is possible to analyze a smaller amount of data to extract the tracklet, obtaining its centroid's position expressed in pixel coordinates therefore

converted to RADEC coordinates.

An additional advantage comes from the fact that the trained neural network achieves better results in terms of correctly detected tracklets on the full-scale original image when compared to the edge-detection algorithm on which ASTRiDE is based. This is valid both for the algorithm based on Scikit-image processing and for the algorithm based on Matplotlib.pyplot processing. In his thesis [27], Calvi compared the detection accuracy of these three methods on a set of chosen images and showed that ASTRiDE can correctly detect about 65% of the present tracklets, the neural network based on skimage processing achieves 98% of correct detection, while the neural network based on matplotlib processing achieves an accuracy of 91%. Comparing the two CNN architectures, the most accurate configuration is also the slowest one, needing about 7 seconds per image for processing and detection steps. On the other hand, the one with lower accuracy needs only about 0.8 seconds to process an image and detect the tracklet. These timing results reported in Calvi's work strongly depend on computer hardware. In both cases, it can be concluded that the detection algorithm based on ML can correctly recognize a higher number of tracklets with respect to ASTRiDE. The latter can then be facilitated in the extraction process if it has to analyze a smaller portion of the image within which the tracklet has already been provided by a more robust detection tool.

Image cropping

A smaller portion of the image is obtained by cropping the original one around the bounding box obtained in output from the RID tool.

The input parameters for the cropping algorithm to work are:

- c_x and c_y , center coordinates of the bounding box, expressed as a floating point value ranging from 0 to 1 (see columns 2 and 3 of Fig. 2.11b);
- w_x and w_y , width and height of the bounding box, expressed as a floating point value ranging from 0 to 1 (see columns 4 and 5 of Fig. 2.11b);
- *dimension*, image dimension along one image axis expressed as the number of pixels;
- *crop_factor*, cropping factor expressed as a decimal value ranging from 0 to 1;
- *margin*, safety margin to ensure the tracklet is fully enclosed inside the cropped image expressed as number of pixels.

By choosing a *crop_factor* equal to 0.5 and working with an image in FITS format having 4096 pixels per axis, the resulting cropped image will have half the original dimension on each axis, resulting in a 2048 by 2048 pixels image. The same *crop_factor* equal to

0.5, applied to the same image in PNG format, which has the same dimensions but a lower resolution of 1024 pixels per axis (or 512, depending on the image processing adopted), leads to a cropped image of 512 by 512 pixels (or 256 by 256, depending on image processing). What changes when the same *crop_factor* is applied to the same image in two different formats is not the image size, but its resolution, due to the down-scaling performed in the image processing and conversion step. On the other hand, maintaining the same image format and cropping with a *crop_factor* equal to 0.5 results in a cropped image with half the number of pixels on each axis, not changing resolution but image size. While performing the cropping step, it must be ensured that the required *crop_factor* is not too small for the tracklet enclosed by the bounding box to be entirely contained within the cropped image. If this happens, the initially selected *crop_factor* is automatically increased. Another check to be performed consists in verifying that the size of the cropped image is consistent with the *crop_factor* selected, with the borders of the cropped image not extending beyond the borders of the full-size image. If this happens, the borders are forced to be coincident; in this case, the tracklet will not be centered inside the cropped image, but it will be shifted towards the borders.

The main structure of the cropping algorithm is reported in Alg. 3.1.

Algorithm 3.1 Image cropping

```

1: %% Obtain the dimension of the cropped image, expressed as number of pixels
2:  $dim\_cut = \text{ceil}(crop\_factor * dimension)$ 
3: %% Obtain the parameters  $c_x$ ,  $c_y$ ,  $w_x$  and  $w_y$  expressed as integer values
4:  $c_x = \text{round}(c_x * dimension)$        $c_y = \text{round}(c_y * dimension)$ 
5:  $w_x = \text{ceil}(w_x * dimension)$        $w_y = \text{ceil}(w_y * dimension)$ 
6: %% Check that the bounding box enclosing the tracklet is fully contained in the
   cropped image. If not, increase the size of the cropped image
7: if  $w_x + margin > dim\_cut$  or  $w_y + margin > dim\_cut$  then
8:    $new\_crop\_factor = \text{ceil}((\max(w_x, w_y) + margin) / dimension)$ 
9:    $dim\_cut = \text{ceil}(new\_crop\_factor * dimension)$ 
10: end if
11: %% Define the borders of the cropped image
12:  $x_l = c_x - dim\_cut / 2$        $x_r = c_x + dim\_cut / 2$ 
13:  $y_l = c_y - dim\_cut / 2$        $y_r = c_y + dim\_cut / 2$ 
14: %% Check that the requested crop does not extend beyond the edges of the original
   image. If it does, change borders of the cropped images
15: if  $x_l < 0$  then
16:    $x_l = 0$ 
17:    $x_r = dim\_cut$ 
18: else if  $x_r > dimension$  then
19:    $x_r = dimension$ 
20:    $x_l = dimension - dim\_cut$ 
21: end if
22: if  $y_l < 0$  then
23:    $y_l = 0$ 
24:    $y_r = dim\_cut$ 
25: else if  $y_r > dimension$  then
26:    $y_r = dimension$ 
27:    $y_l = dimension - dim\_cut$ 
28: end if
29: %% Perform the effective crop
30:  $cropped\_image = original\_image[y_l : y_r + 1, x_l : x_r + 1]$ 

```

3.2. Analysis of images converted from FITS to PNG format

In this first approach, the cropped images are those converted into PNG format by the RID tool, before tracklet detection takes place. After being cropped around the bounding box enclosing the tracklet, images in PNG format are given as input to the astrometric calibration tool (*Astrometry.net*). This tool, once obtained the WCS coordinates, generates a new FITS format file, which is the input for the tracklet extraction process. During the processing and conversion step, what was written in the header of the original FITS file is lost. The original header must then be restored into the new FITS file that is generated after the astrometric calibration process. Some pieces of information inside the original header, such as the observation date, the exposure time, and the object's NORAD ID, are indeed necessary for determining the tracklet's centroid RADEC coordinates, computing the time bias, and checking the accuracy of the pipeline.

The most notable feature of this approach lies in the loss of resolution associated with the image conversion from FITS to PNG format. Besides being down-scaled, images are also down-converted from 16-bit to 8-bit pixel depth. Image format conversion is therefore a double source of loss of information. This makes the images smaller in terms of file size and makes both the astrometric calibration process and the tracklet extraction process perform more quickly. Another possible advantage of this configuration is the fact that the images in PNG format are subject to image processing, which emphasizes the light sources and thus makes it easier for the image to be associated with the correct portion of the sky during the astrometric calibration process and for ASTRiDE to extract the tracklet. This could result in a high percentage of images on which the entire pipeline process can be completed. On the other hand, a possible and relevant drawback caused by the loss of resolution is the loss of precision in obtaining the position of the tracklet's centroid and therefore of the RADEC coordinates of the object detected in the image.

The block diagram representing this approach is shown in Fig. 3.2

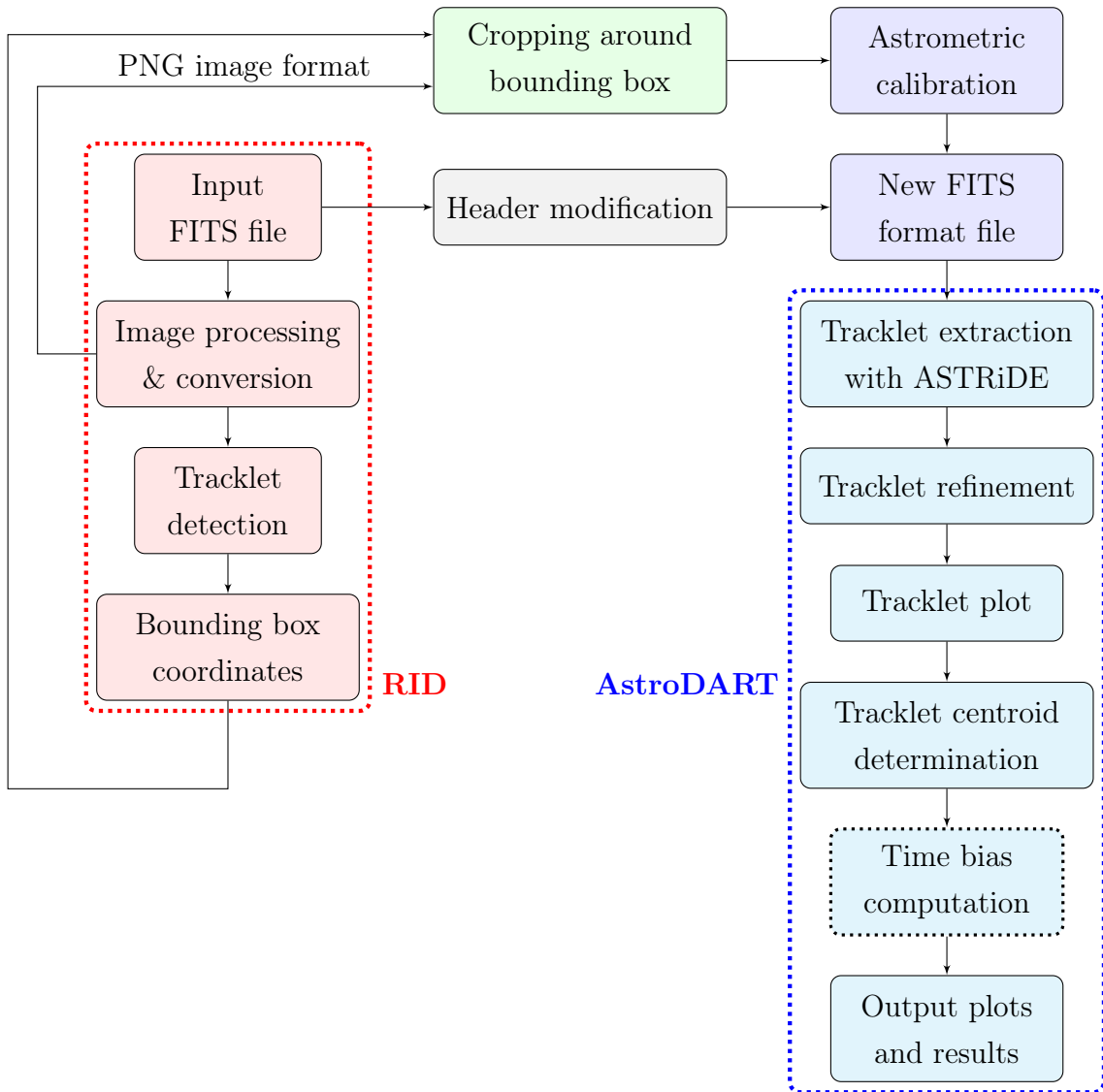


Figure 3.2: Analysis of images in PNG format

Regarding the RID tool, image processing and tracklet detection steps can be performed with two different strategies, each one based on a different processing and a different CNN architecture. The first strategy includes image processing based on the Scikit-image library and a more complex neural network; this image processing is slow but leads to high-quality images with easily identifiable streaks. The second strategy is based on image processing implemented with Matplotlib.pyplot library and is characterized by less complex neural network architecture, aiming at a faster even if less accurate tracklet identification. Both possibilities are considered for the analysis of the overall pipeline.

3.3. Analysis of images in FITS format

The second approach is based on the analysis of the cropped images in FITS format. In this case, the tracklet detection algorithm is used only to obtain the geometric coordinates of the bounding box, while the image to be cropped is the original one in FITS format. This allows keeping the original resolution and the original pixel color depth intact, varying only the size of the image undergoing the tracklet extraction process, depending on the chosen crop factor. Maintaining the resolution and avoiding loss of information allows to obtain higher accuracy in the estimation of the centroid's position and therefore of the object's RADEC coordinates. On the other hand, the only reduction in file size is that associated with image cropping. This makes the overall analysis more time-consuming, compared to the case in which images in PNG format are analyzed.

A block diagram showing this approach is reported in Fig. 3.3

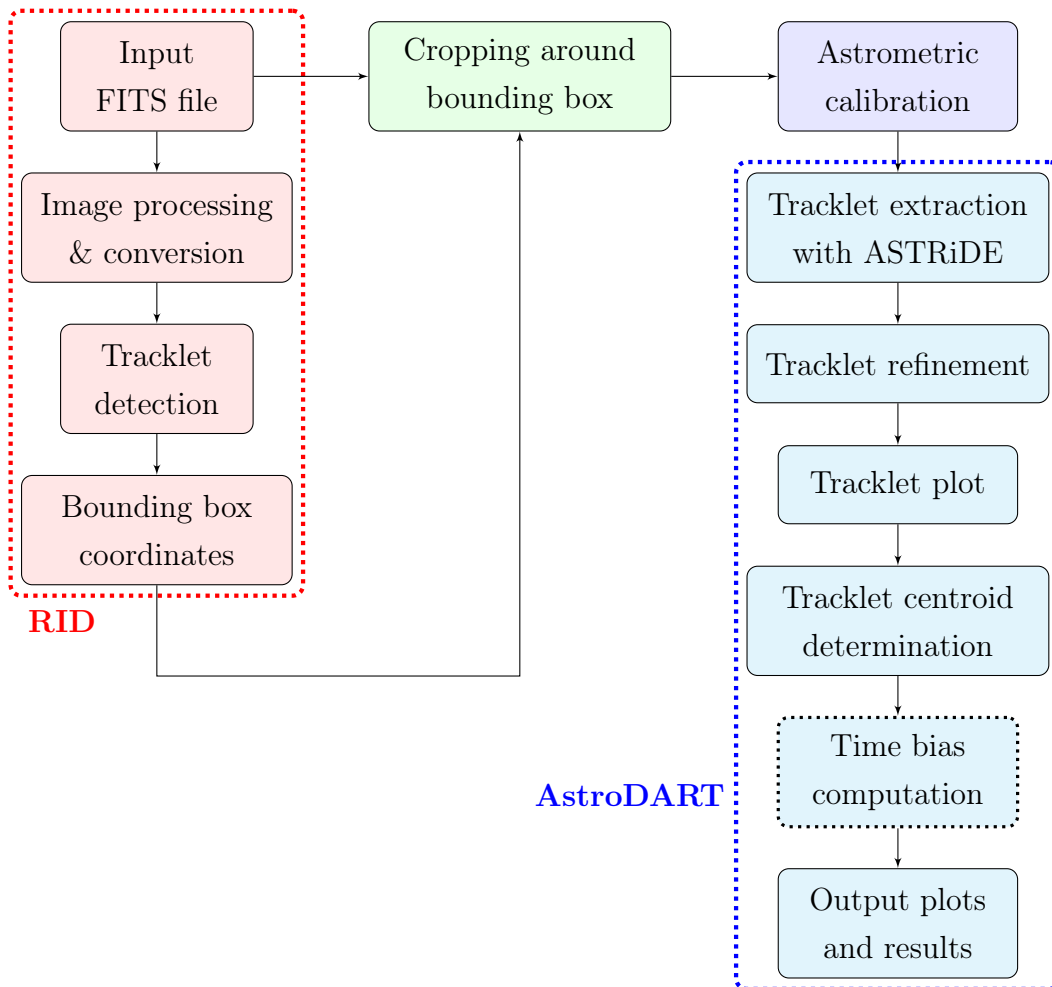


Figure 3.3: Analysis of images in FITS format

For each image, different crop sizes are considered, to evaluate how the computational time, the success rate of the overall pipeline, and the accuracy in determining the RADEC centroid coordinates vary with the size variation of the image undergoing the tracklet extraction process. The total computational time is given by the sum of three factors:

1. time required for image processing and tracklet detection by RID;
2. time necessary to perform image astrometric reduction;
3. time needed for tracklet extraction and the following steps implemented in the AstroDART tool.

The time needed to crop the image is neglected, as well as the time needed to modify the header of the new FITS file generated after the astrometric reduction, in case this step is required.

The overall pipeline run is considered to be successful if the analyzed image is solved during the astrometric calibration step and if a single tracklet is identified and extracted by ASTRiDE, allowing the refinement process and the determination of the RADEC coordinates of the tracklet's critical points (most importantly of the centroid).

The accuracy in obtaining the RADEC coordinates of the tracklet's centroid from the cropped images is firstly evaluated taking the estimation of the same RADEC coordinates, obtained from the full-scale FITS format image, as reference. Then, if the RADEC coordinates obtained from the propagation of the Two-Line Elements (TLE) catalog data are available, together with the time bias data, the centroid RADEC residuals after time bias correction are also taken as reference for accuracy evaluation.

4 | Results

This chapter is devoted to the presentation of the results obtained by introducing object detection and image cropping in the pipeline for tracklet extraction, and by varying the size of the cropped images. Section 4.1 shows the results obtained by analyzing the cropped images in PNG format, while Sect. 4.2 presents the results of the analysis of cropped images in FITS format.

The presented results are obtained from algorithms based on Python codes executed on a 2016 machine with an i7-6700HQ quad-core and eight-threads CPU (max frequency 2.6GHz, using Turbo Boost 3.5GHz), 16Gb of RAM, and an Nvidia GTX 950M GPU with 4Gb of VRAM. The operating system is Ubuntu 20.04.5 LTS.

4.1. Pipeline performances analyzing images in PNG format

In order to perform the analysis, an initial set of 94 images has been filtered, keeping only those images whose analysis is completed by the original AstroDART pipeline, resulting in 82 images obtained from 5 different observation campaigns. This filtering has the disadvantage of partially hiding both the difference in terms of computational time between a non-filtered set of full-scale FITS images and the same images cropped and converted to PNG format and the difference in terms of the overall success rate of the pipeline. On the other hand, the advantage is that on these filtered images it is possible to compare the accuracy in determining the RADEC tracklet centroid coordinates.

All these images are labeled with the NORAD ID of the observed object and contain at least one tracklet, with 2 images containing 2 tracklets each.

In Tab. 4.1 relevant information about each of these groups of images is reported.

Campaign number	Number of images	NORAD ID	Observation date (UTC)
2756	12	32376	2017-09-15 18:17:58
2761	19	32376	2017-09-20 18:11:44
2762	13	29499	2017-09-20 19:21:28
2763	15	32376	2017-09-21 03:30:54
2764	23	37216	2017-09-21 03:53:58

Table 4.1: Observation campaigns details

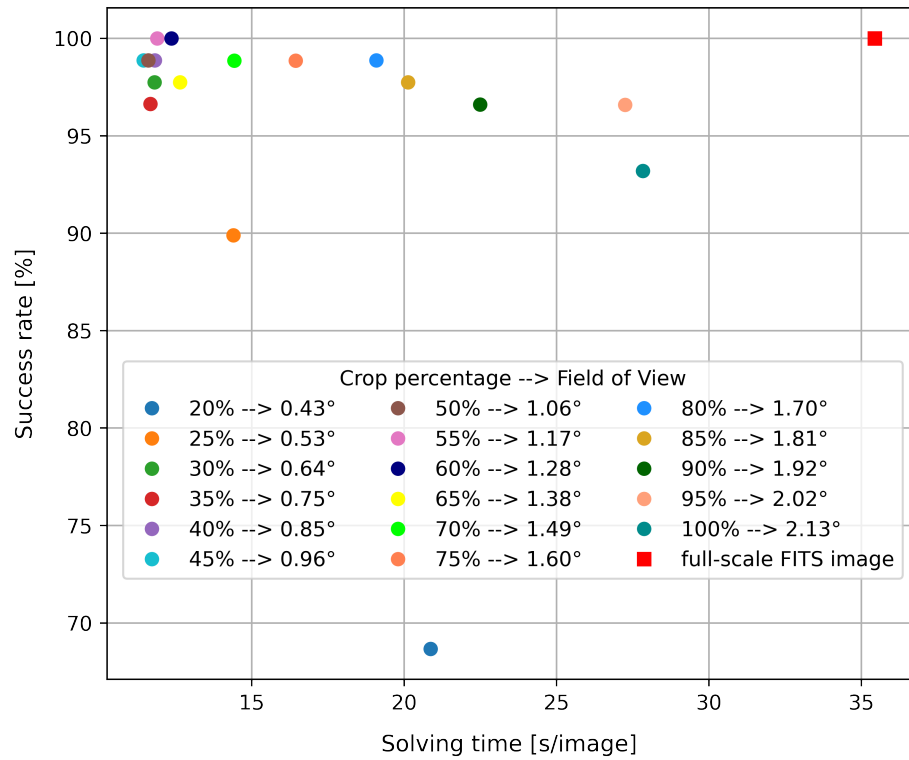
Regarding the ASTRiDE parameters which rule the tracklet search at the beginning of the AstroDART pipeline, when FITS format images are analyzed, they are leaved unchanged as they were set in the original pipeline [30]: *contour_threshold* = 2, *shape_cut* = 0.1, *radius_dev_cut* = 0.5 and *area_cut* = 1000. The *area_cut* parameter is decreased to 200 when PNG format images are analyzed, while the other parameters remain unchanged.

Scikit-image processing: success rate and computational cost

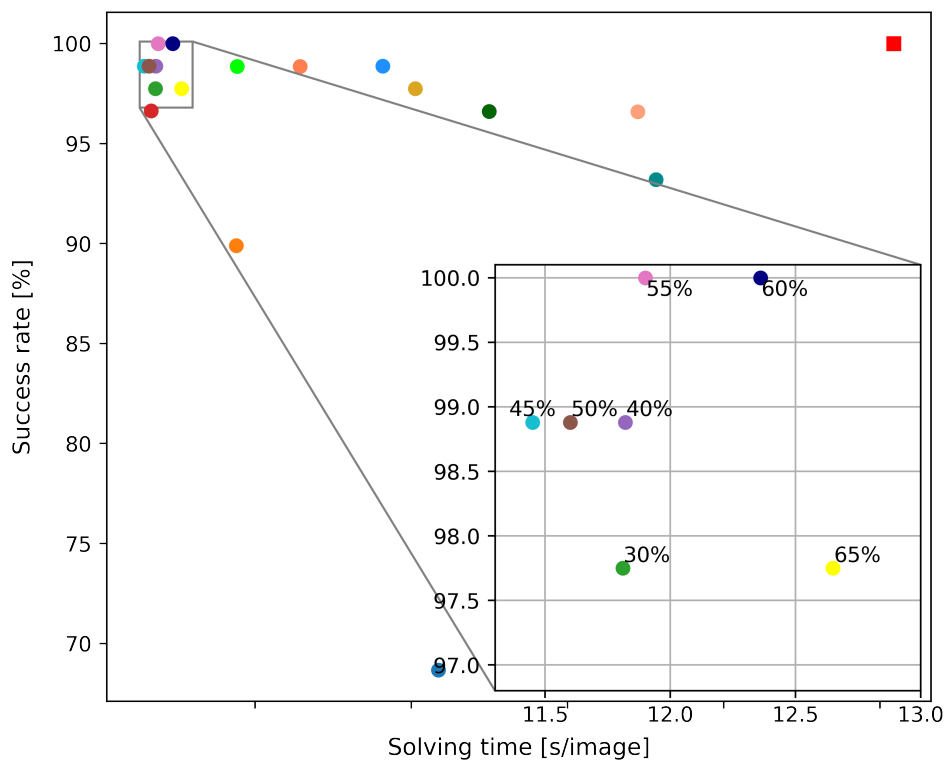
Figure 4.1a shows how the first two parameters of merit chosen for this analysis, i.e. the pipeline success rate and the cost in terms of computational time required, vary with the variation of the size of the analyzed images in PNG format.

Each point in the graphs represents images containing a tracklet and cropped around it with a given crop factor (crop percentage is reported in the legend); each crop factor is associated with a value of the FoV framed in the image. The two graphs shown in Fig. 4.1 represent the same trend of these points, with the only difference being that in Fig. 4.1b the plot region containing the most optimal points, which are associated with a high success rate and a low computational time, is zoomed in. The red squared right-most point represents images in their original FITS format. Numerical values from which this graph is obtained can be found in Tab. A.1, where computational times and success rates associated with the most relevant steps of the pipeline can be distinguished.

Looking at these results, it can be stated that the format conversion and the cropping that results in images with about 1° of FoV allows a much faster analysis, keeping a very high success rate of the analysis itself. The lower limit on the crop factor value is given by the tracklet’s length, with this control automatically implemented, as shown in Alg. 3.1. Even if the tracklet was entirely contained in a cropped image with a restrictive crop factor, the probability that this image will not be recognized by the astrometric calibration tool increases, as it is likely that fewer sources (stars) would be present in this image, making more difficult its recognition during the astrometric calibration step.



(a)



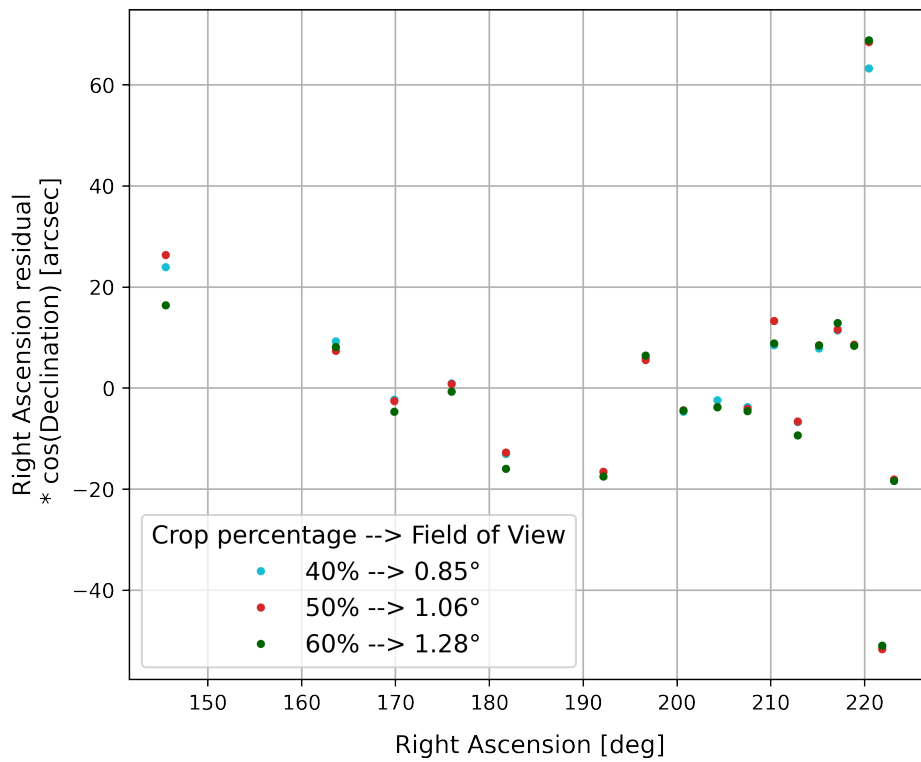
(b)

Figure 4.1: Effect of size variation of PNG images on pipeline success rate and total computational time (with Scikit-image processing)

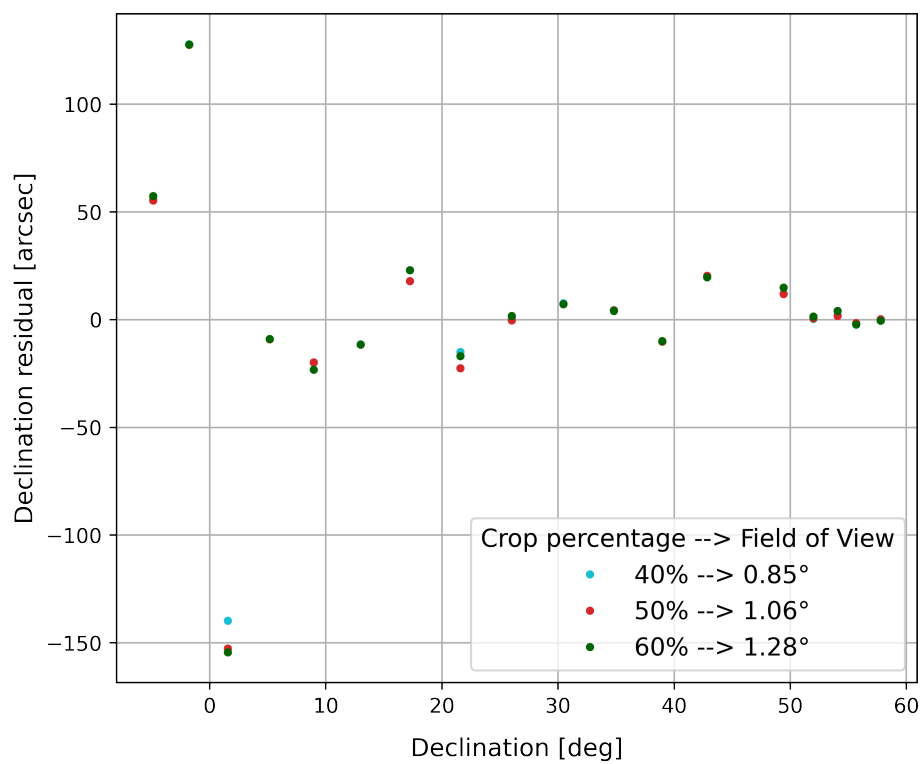
The results shown in Fig. 4.1a are obtained using the strategy with slower image processing (Scikit-image processing) and more complex CNN architecture within the RID tool, leading to higher image quality; the time needed for image processing and tracklet detection is about 6 seconds per image, having exploited the use of the Graphics Processing Unit (GPU). Using the Central Processing Unit (CPU), the time required for detection is slightly higher, about 8 seconds per image. A further advantage in terms of computational cost is obtained thanks to the use of the Ubuntu operating system, both regarding the astrometric calibration process, which otherwise should be performed through the use of the Virtual Machine (VM) and regarding the tracklet detection process with RID. Regarding the latter step, adopting the strategy with the most accurate processing (Scikit-image processing), the use of the Windows operating system would imply a detection time of about 17 seconds per image with the use of the CPU and about 12 seconds per image with the use of the GPU. From these data and from the data available from Calvi's work [27], in which the use of the Windows operating system is known, it can be deduced that the time required for tracklet detection could be lowered below 6 seconds per image with the use of higher-quality hardware.

Scikit-image processing: accuracy in determining RADEC centroid coordinates

After evaluating the performance of the pipeline based on the first two parameters of merit, now it is evaluated how the precision in determining the position of the tracklet centroid and therefore its RADEC coordinates varies with the image cropping, keeping the format conversion. For this purpose, the RADEC coordinates of the tracklet centroid obtained through the analysis of the full-scale FITS images carried out through the original AstroDART pipeline are taken as a reference. The residuals are then calculated both in terms of Right Ascension and Declination, as the difference between the value obtained in output from the analysis of the cropped image in PNG format and the value obtained from the analysis of the full-scale image in FITS format. This is not done for each value of the crop factor shown in the previous graph, but only for a few optimal values that involve a FoV of the cropped image of about 1° ; specifically, the results obtained by cropping with crop factor values equal to 0.4, 0.5 and 0.6 are shown. Figure 4.2a and Fig. 4.2b show the case related to a subset of images coming from a single data acquisition campaign (campaign number 2761 in Tab. 4.1). The results obtained from the images of the other four campaigns are shown in App. B (Fig. B.1 to Fig. B.4).



(a)



(b)

Figure 4.2: Centroid RADEC residuals (Scikit-image processing) - acquisition campaign number 2761

The Right Ascension residuals are found by multiplying the difference in angles with a factor given by the cosine of the Declination. The reason is due to the geometrical features of spherical triangles: a small angular error close to the tracklet may turn out to be quite larger in terms of Right Ascension. Conversely, a large residual in the Right Ascension can correspond to a small angular error on the tracklet. This comes from the fact that Right Ascension is defined on the Earth's equator at Declination = 0° . The way to correct this phenomenon is to consider $RA\cos(DEC)$ to evaluate the residuals.

Observing the results shown in Fig. 4.2a and Fig. 4.2b, it is noted that it is almost always present a difference that can reach few tens of arcseconds between the RADEC coordinates obtained from the analysis of the cropped images in PNG format and the same coordinates obtained from the analysis of full-scale images in FITS format. In some cases, these differences are even bigger in terms of arcseconds, but in such circumstances this is often due to a different identification of the tracklet by ASTRiDE. An example of such a case is given in Fig. 4.3

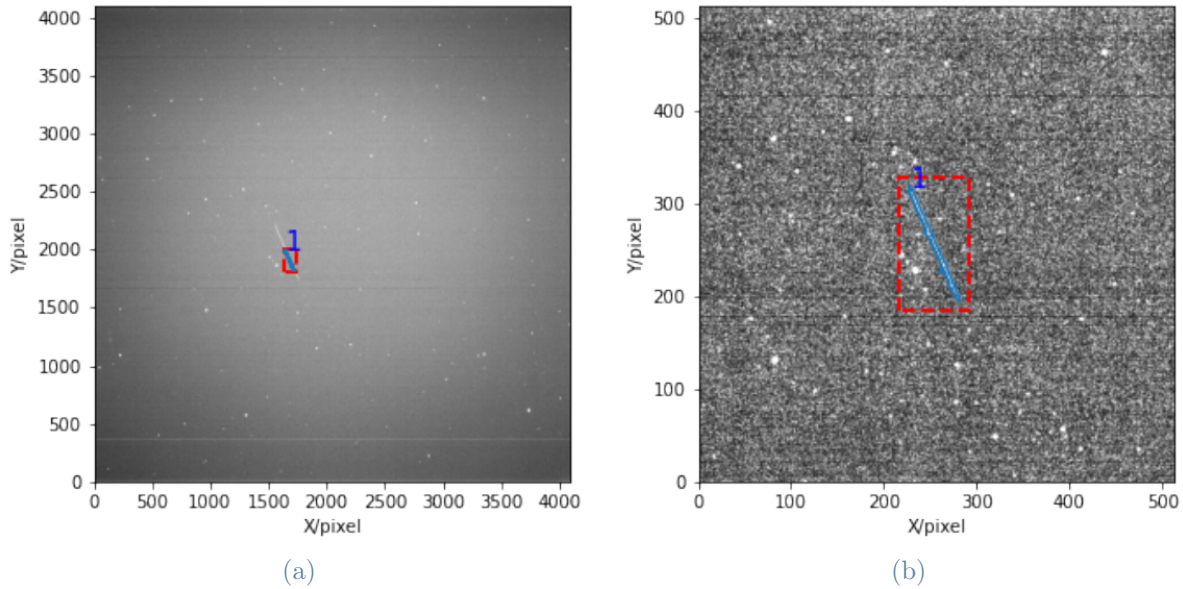


Figure 4.3: Example of different tracklet identification by ASTRiDE

The comparison between the two images shown in Fig. 4.3 refers to the group of points at the top right in Fig. 4.2a and bottom left in Fig. 4.2b. In this case, the high values on the residuals are due to a different identification of the tracklet, in particular to a partial identification in the full-scale image in FITS format (Fig. 4.3a). Those cases in which the residual value on the RADEC coordinates remains limited to a few tens of arcseconds are observed even for a complete tracklet identification both in the full-scale FITS image and in the cropped PNG image.

The observed differences in the determination of the RADEC coordinates of the tracklet's centroid might depend on the loss of resolution associated with the image conversion. Resolution down-scaling can in turn affect both the pointing accuracy during the astrometric calibration step and the tracklet extraction process, with the tracklet's contour points being detected differently with respect to the case in which the full-scale FITS format image is analyzed.

Figure 4.4 shows the difference between the pointing obtained from full-scale images down-scaled in resolution and converted in PNG format and the pointing obtained from full-scale, high-resolution images in FITS format.

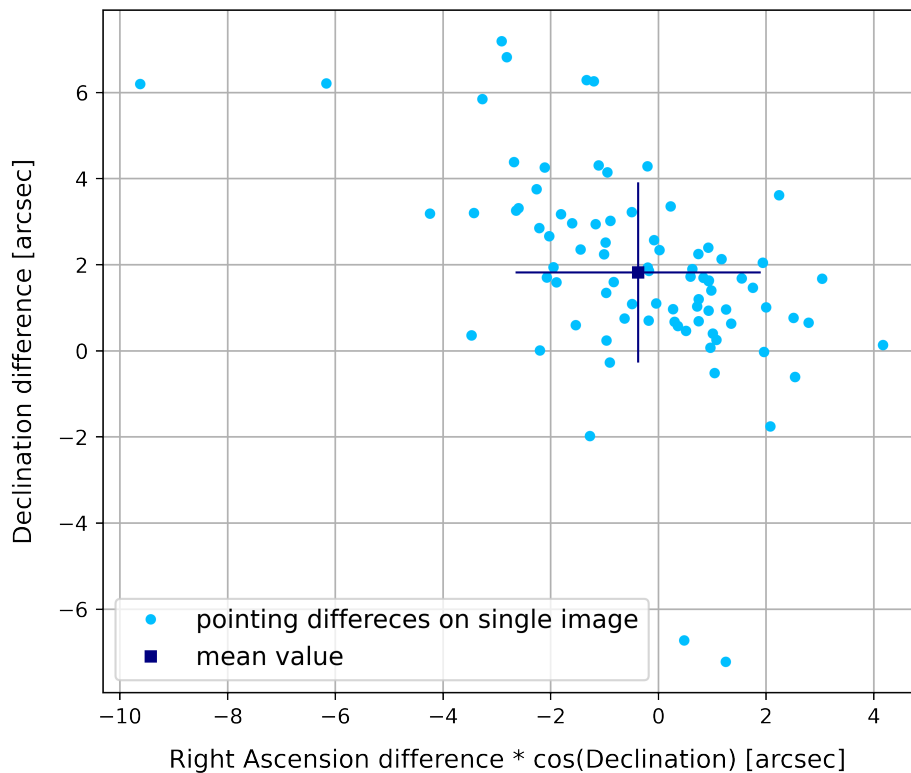


Figure 4.4: Pointing error of images in PNG format (Skimage processing)

These observed differences in pointing due to down-scaled resolution are not enough to justify the differences observed in Fig. 4.2a and Fig. 4.2b. Resolution down-scaling has an impact also in the different determination by ASTRiDE of the points that are part of the tracklet's contour and this affects the calculation of the position of the centroid. An example is shown in Fig. 4.5, where 4.5a shows the refined tracklet as it is extracted from the original FITS image, while Fig. 4.5b shows the same tracklet refined after it is extracted from the image processed with Scikit-image library, converted in PNG format, and cropped with a crop factor equal to 0.5.

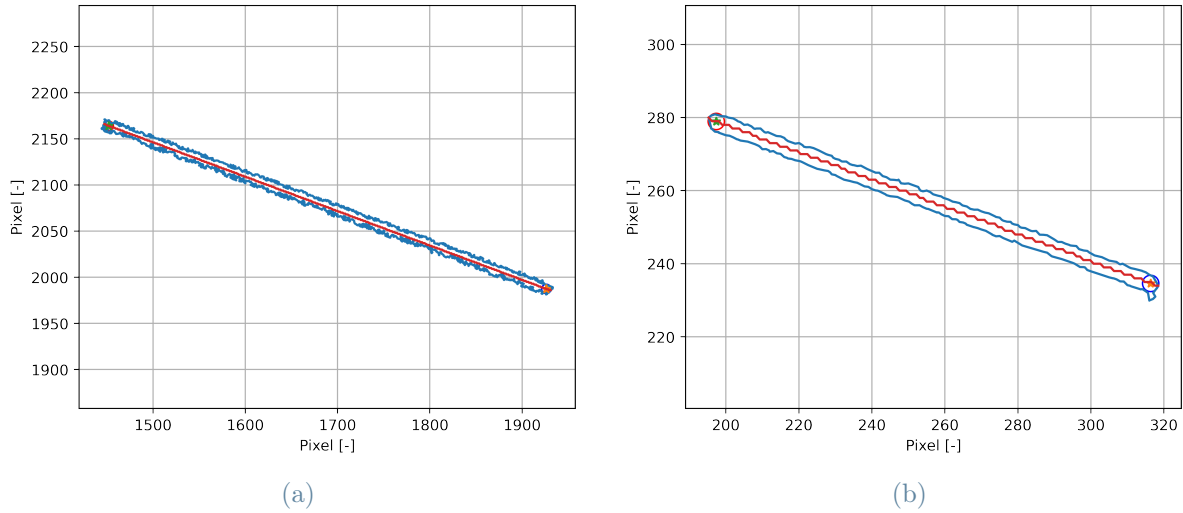


Figure 4.5: Different segmentation after the refinement process - FITS format (a) and PNG format Skimage processing (b)

A second check on the accuracy in determining the RADEC coordinates of the centroid is performed by checking the value of the residuals on the RADEC centroid coordinates calculated as the difference between the coordinates obtained from the image, corrected with the data on the time bias, and those obtained from the propagation of the TLE catalog data. These values are given in output at the end of the AstroDART pipeline. Figure 4.6 compares the residuals obtained from the original full-scale FITS images with those obtained from the cropped PNG images. The images are the same from which the results shown in Fig. 4.2a and Fig. 4.2b were obtained. Also in this case, the results associated with images coming from the remaining four acquisition campaigns are reported in App. B (Fig. B.5 to Fig. B.8).

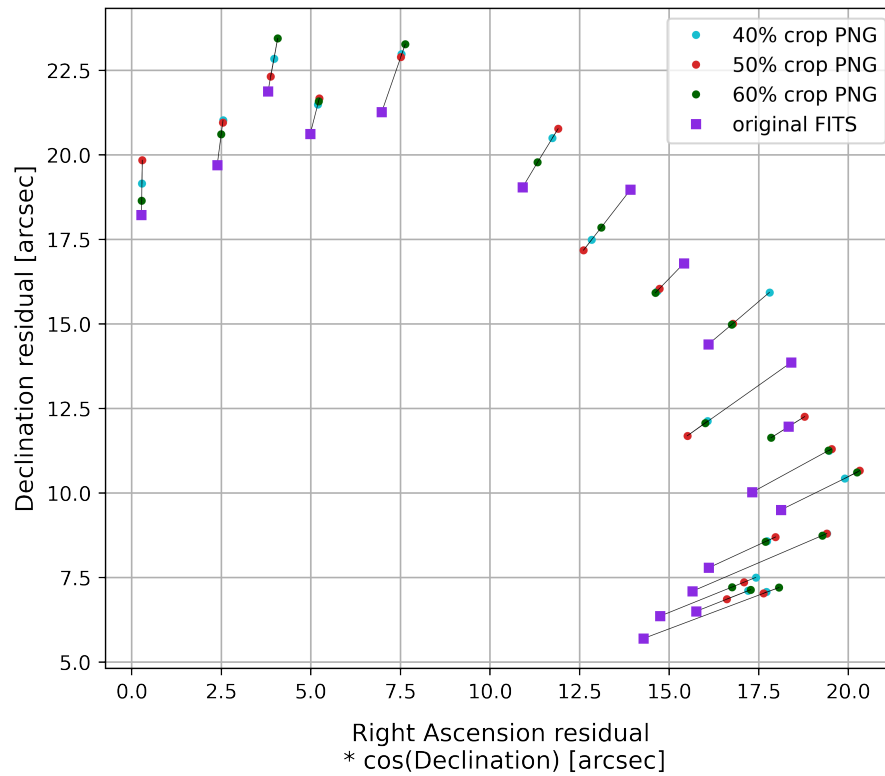
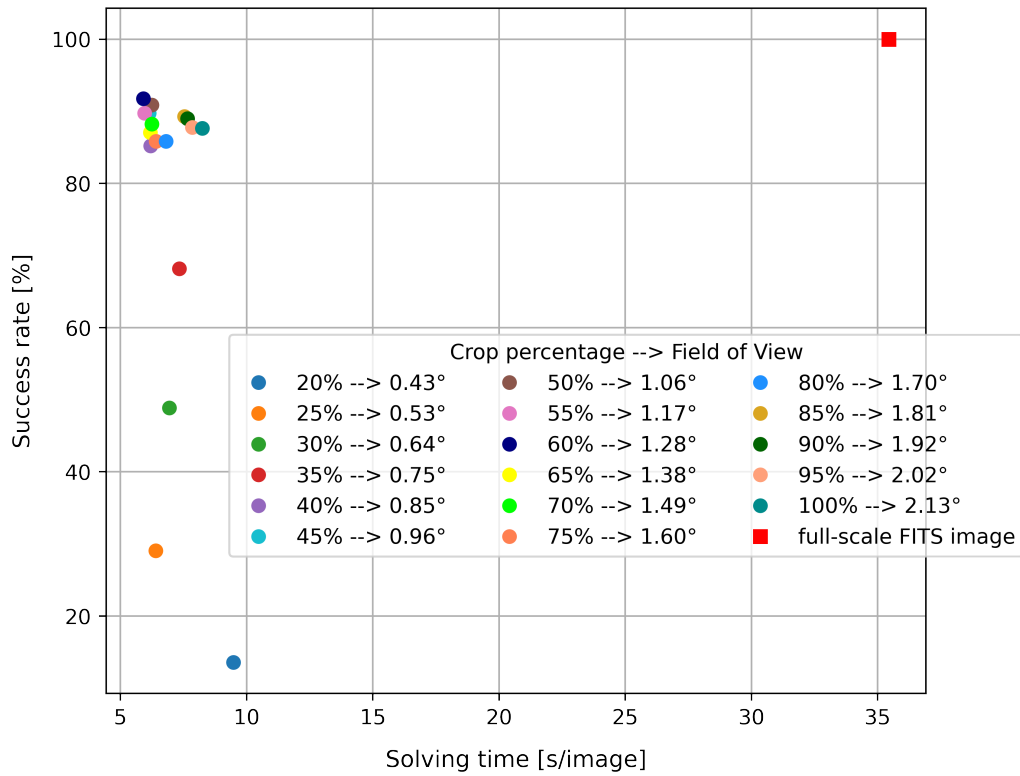


Figure 4.6: Centroid RADEC residuals corrected with time bias (campaign 2761) - Skimage processing

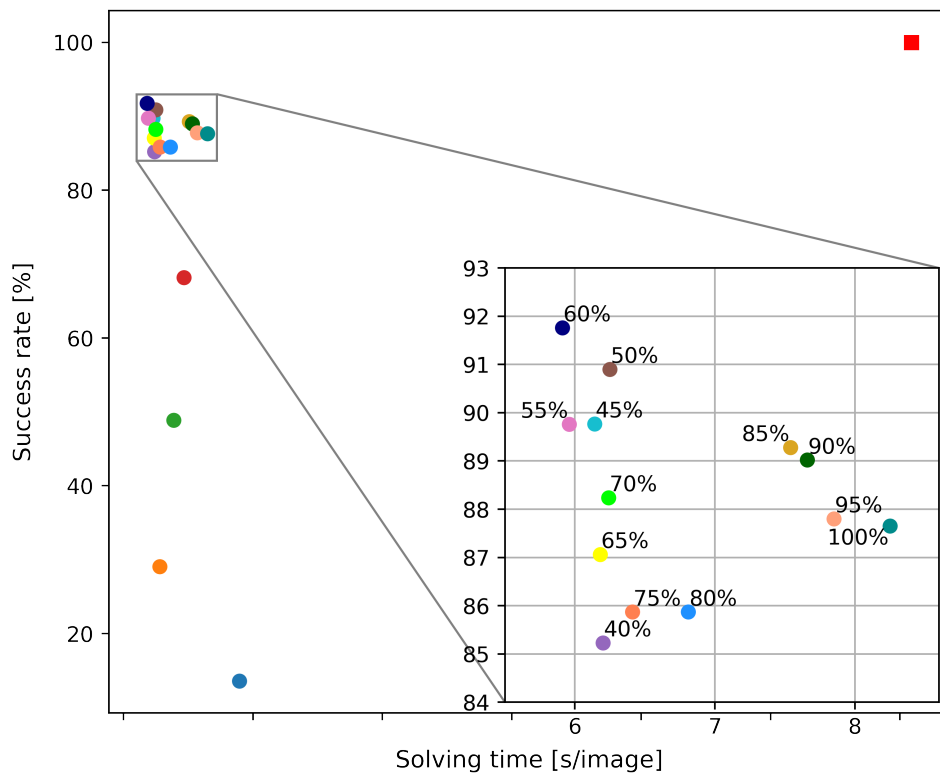
From this graph and also from those reported in App. B, a trend of the points is recognizable, which indicates a worsening in the precision with which the RADEC coordinates of the centroid are obtained when passing from the original full-scale, full-resolution image to the cropped, down-scaled, and converted image.

Matplotlib processing: success rate and computational cost

The next graph shows the results obtained performing the analysis on the same PNG images, adopting in this case the strategy for the tracklet detection step involving a faster but less accurate image processing (Matplotlib processing) and a less complex CNN architecture. Numerical values from which the graph in Fig. 4.7 is obtained can be found in Tab. A.2, where computational times and success rates associated with the most relevant steps of the pipeline can be distinguished.



(a)



(b)

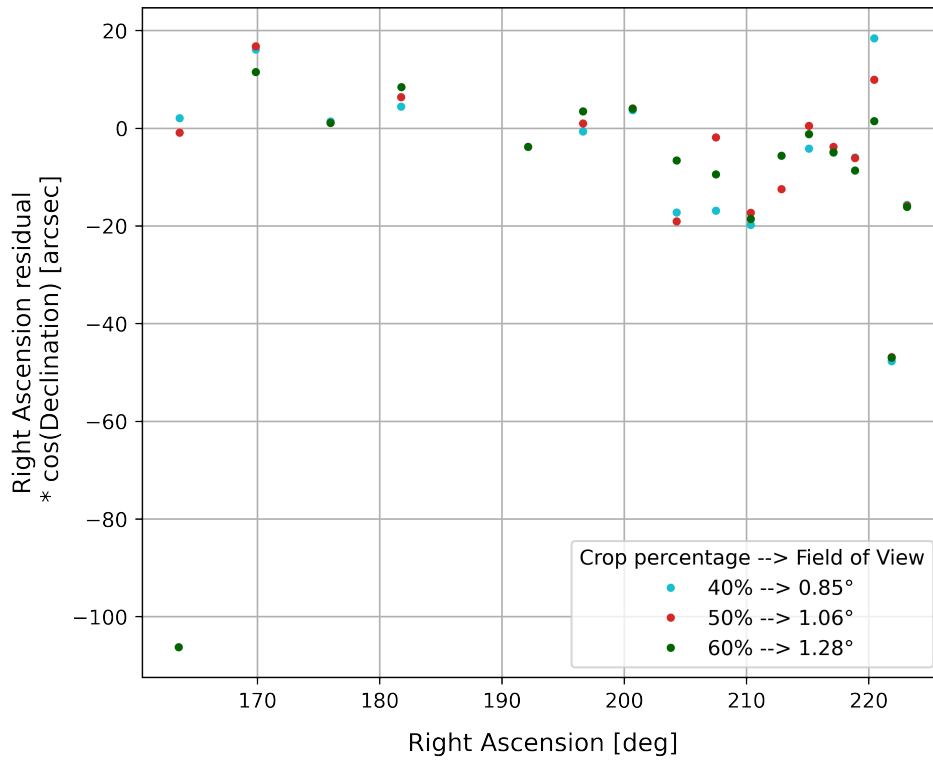
Figure 4.7: Effect of size variation of PNG images on pipeline success rate and total computational time (with Matplotlib.pyplot processing)

Concerning the results shown in Fig. 4.7a and Fig. 4.7b, the most optimal points are again those associated with a framed FoV of about 1° . It can also be noted that there is a larger gap in terms of computational cost between full-scale images in FITS format and cropped images converted to PNG format; in fact, the required time for completing the pipeline drops from 35 to about 6 seconds per image. This is due to the higher resolution loss that reduces the files' size and also to the more rapid image processing and tracklet detection steps, which are performed in only 0.8 seconds per image. The drop in resolution goes from 4096 by 4096 pixels to 512 by 512 pixels. From the graph, it is also observed a reduced ability of the pipeline to complete the analysis, compared to the case shown in Fig. 4.1a. In the case of cropped images with a low crop factor, it is believed that this is due to both the loss of resolution and the lower-quality image processing that cause a loss of the information necessary to detect the light sources (stars), during the astrometric calibration process. In the case of images with higher FoV, this is probably due again to lower quality image processing, which lacks in emphasizing the tracklet within the image and leads to an increasing number of failures during the tracklet extraction process, in particular during the tracklet search with ASTRiDE.

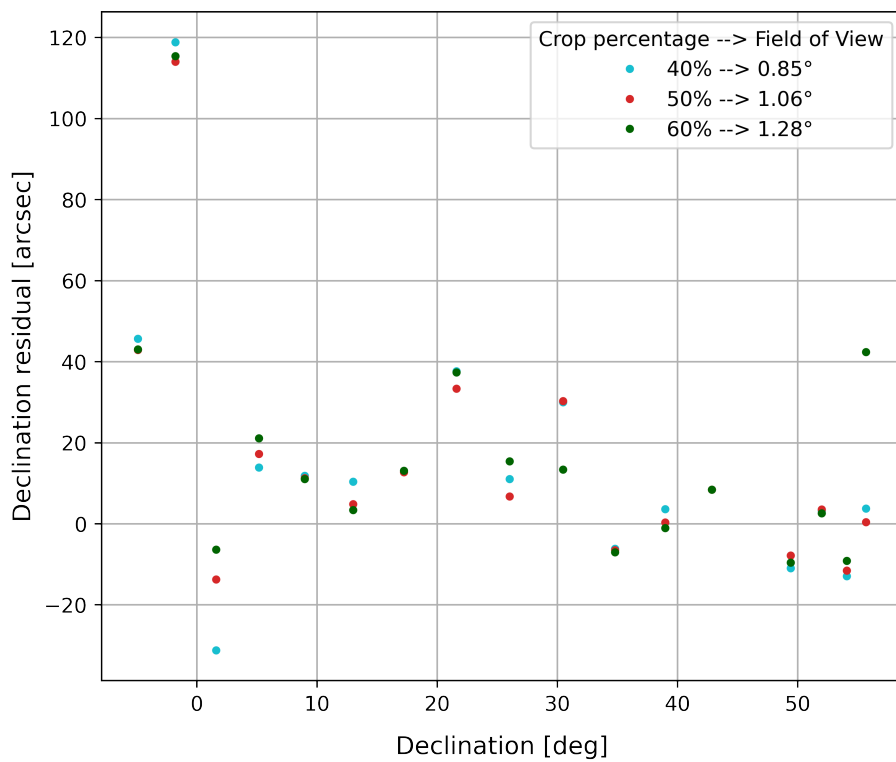
Matplotlib processing: accuracy in determining RADEC centroid coordinates

The precision in determining the position of the tracklet centroid and its RADEC coordinates is evaluated also in this case in which a different image processing is adopted and the resolution is further halved. Figure 4.8a and Fig. 4.8b show the difference between the value obtained in output from the analysis of the cropped image in PNG format and the value obtained from the analysis of the full-scale image in FITS format. The analyzed images are those coming from acquisition campaign number 2761, the same from which the results shown in Fig. 4.2a and Fig. 4.2b were obtained.

Also in this case, the observed differences are in the order of a few tens of arcseconds, with outlier points indicating a difference in tracklet identification, like in the example shown in Fig. 4.3.



(a)



(b)

Figure 4.8: Centroid RADEC residuals (Matplotlib processing) - acquisition campaign number 2761

The differences in the values of the centroid RADEC coordinates between those obtained from PNG cropped images and those obtained from the analysis of the whole image in FITS format appear slightly higher than in the previous case, especially looking at the data from the other four data acquisition campaigns, reported in App. B (Fig. B.9 to Fig. B.12). This is consistent with the further reduction in resolution, which also leads to a larger difference between pointing obtained from full-scale PNG images and that obtained from full-scale FITS images during the astrometric calibration process, as can be seen in Fig. 4.9

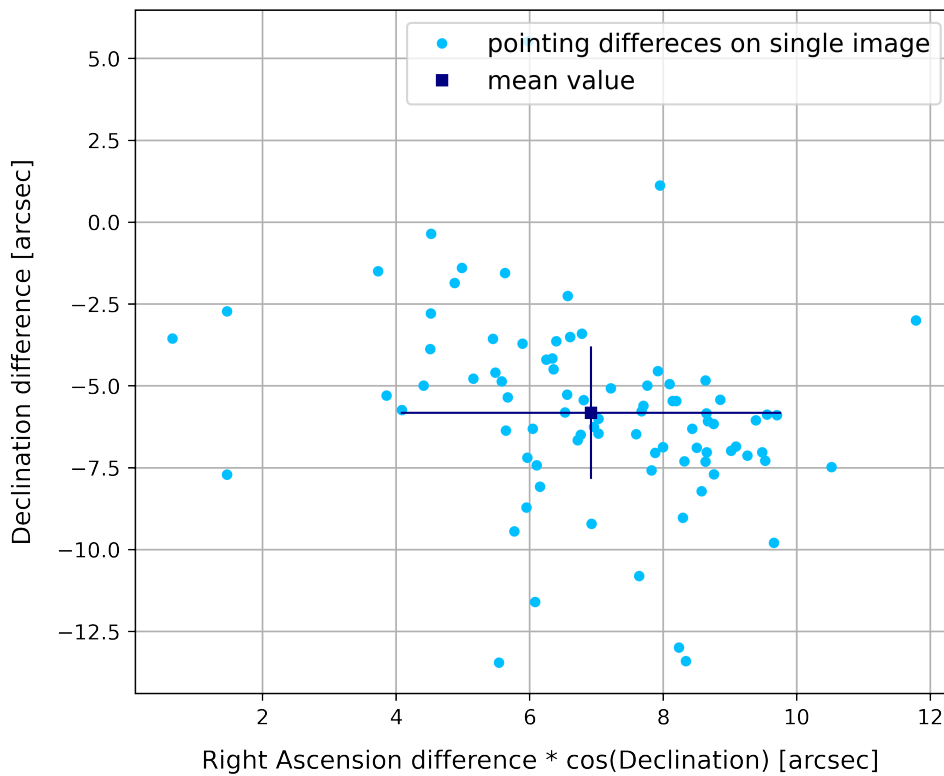


Figure 4.9: Pointing error of images in PNG format (Matplotlib processing)

This increase in pointing difference can be explained by the increase in the $deg/pixel$ ratio of the images having now a resolution of 512 by 512 pixels. Table 4.2 shows the trend of the $deg/pixel$ ratio as the image resolution varies.

Image format	Image processing	Resolution [<i>pixels</i> × <i>pixels</i>]	<i>deg/pixel</i> ratio
FITS	-	4096 × 4096	1.87
PNG	Scikit-image	1024 × 1024	7.48
PNG	Matplotlib	512 × 512	14.7

Table 4.2: *deg/pixel* ratio variation with resolution

The difference in the RADEC centroid coordinates illustrated in Fig. 4.8a and Fig. 4.8b is only partially due to the shift in pointing. The other cause is a different segmentation of the tracklet by ASTRiDE, in the process of tracklet extraction. An example is given in Fig. 4.10, where the same tracklet shown in Fig. 4.5 is reported, with the only difference that now the comparison is made between the tracklet from the original FITS image and the one from the image processed with the Matplotlib library, converted in PNG format, and cropped with a crop factor equal to 0.5.

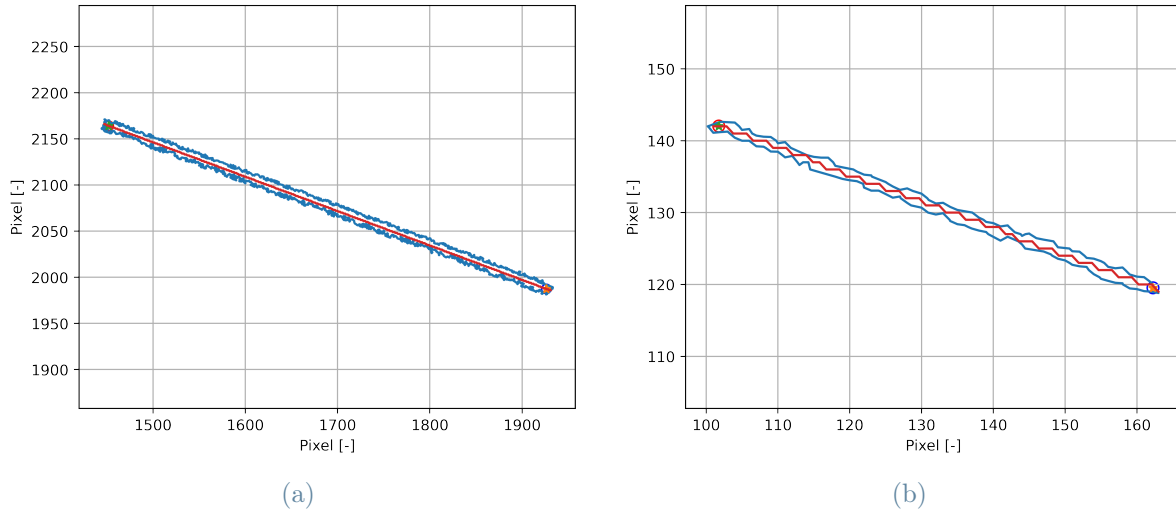


Figure 4.10: Different segmentation after the refinement process - FITS format (a) and PNG format Matplotlib processing (b)

Finally, the check on RADEC centroid time-bias-corrected residuals calculated as the difference with respect to propagated TLE data is performed. Figure 4.11 compares the time-bias-corrected residuals obtained from the original full-scale FITS images with those obtained from the cropped PNG images processed with the Matplotlib library.

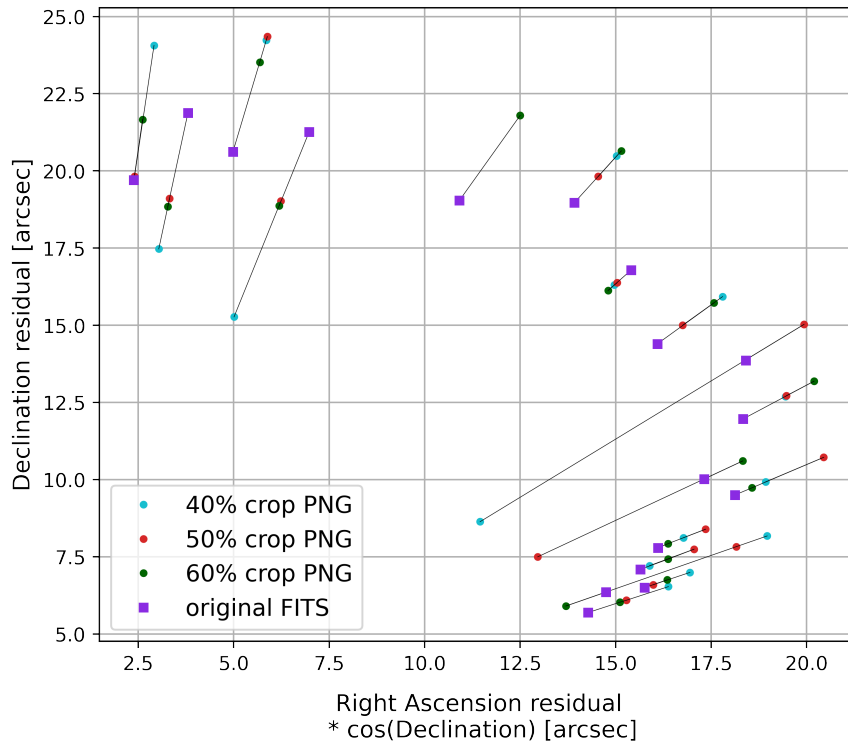


Figure 4.11: Centroid RADEC residuals corrected with time bias (campaign 2761) - Matplotlib processing

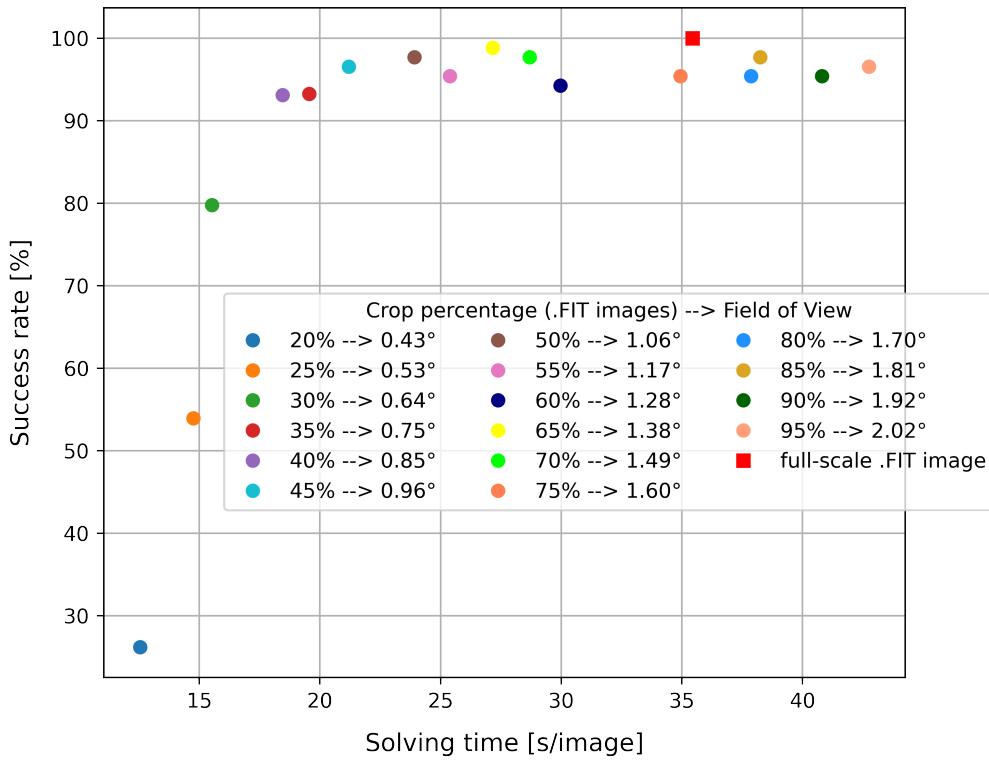
The trend is similar to that shown in Fig. 4.6, with a general worsening of residuals when passing to images in PNG format. The only difference is that in this case, more marked differences are observed between the cropped PNG images and the original FITS images, as can be seen also in images from the other campaigns, reported in App. B (Fig. B.13 to Fig. B.16).

4.2. Pipeline performances analyzing images in FITS format

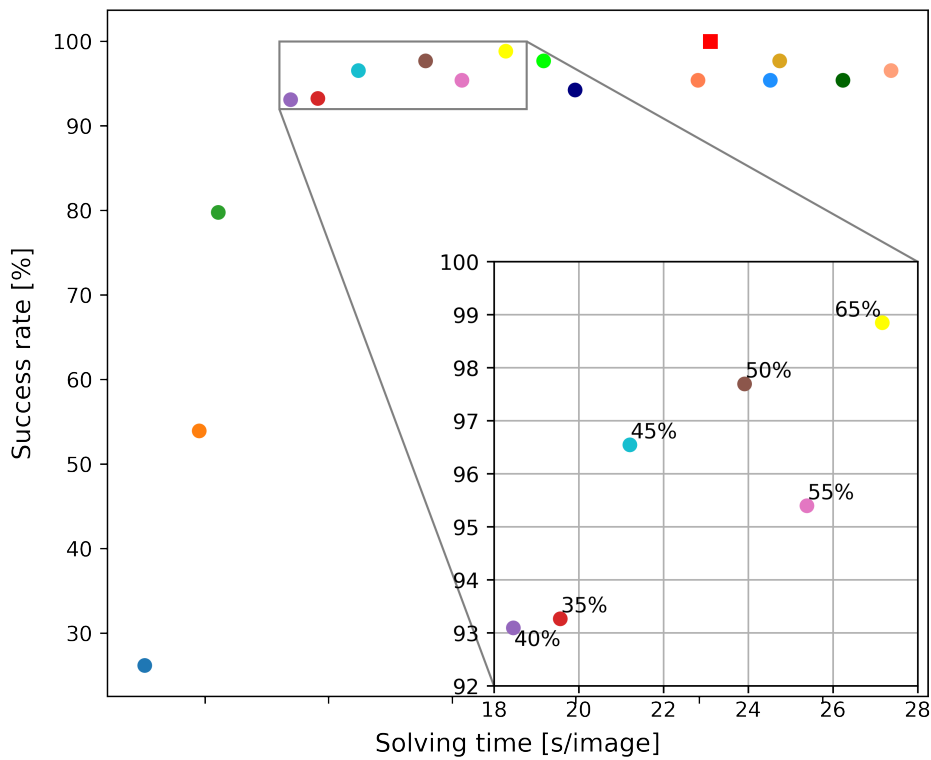
In this section, the results obtained by performing the tracklet extraction process on the cropped image in FITS format are reported. In this case, the original high-resolution is preserved through the pipeline.

Success rate and computational cost

Figure 4.12a shows the trend of the pipeline success rate and of the required computational time with the dimensions variation of high-resolution FITS images.



(a)



(b)

Figure 4.12: Effect of FITS images size variation on pipeline success rate and total computational time

Numerical values from which the graph is obtained can be found in Tab. A.3. The tracklet detection step is performed adopting the slower but higher-quality image processing (through Scikit-image library) and the more complex CNN architecture, and hence needs about 6 seconds per image to be completed.

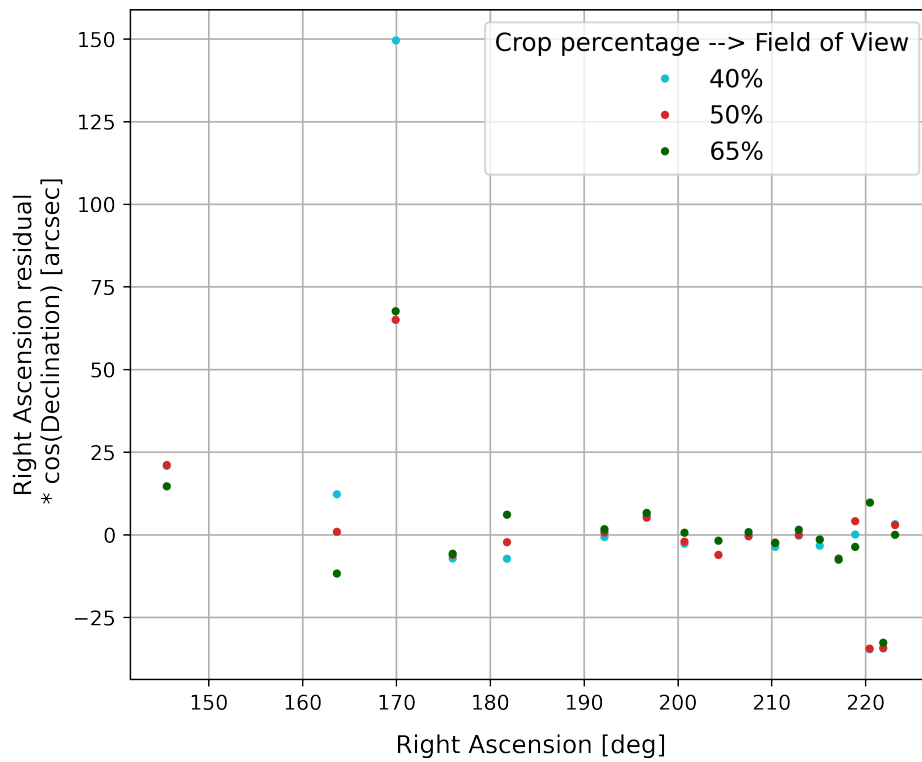
Observing the graph in Fig. 4.12 and comparing it with the analogous ones shown in Fig. 4.1 and Fig. 4.7, it is possible to notice greater difficulty in identifying optimal points, although the best results, which guarantee a high percentage of success rate of the pipeline and a computational cost not particularly high, they are still obtained for cropped images leading to frame a FoV of about 1° .

It is also noted that the cropping of the images brings a lower computational cost saving compared to the cases in which images in PNG format are analyzed; this is because maintaining the high resolution through the pipeline flow leads to a larger size of the files to be analyzed. Considering a crop factor equal to 0.5 applied to the initial FITS images, there is a saving in computational time of about 30%. It should also be remembered that this computational cost saving is underestimated due to the initial selection of the analyzed images, of which it is known a priori that they are successfully analyzed by the original pipeline of the AstroDART tool.

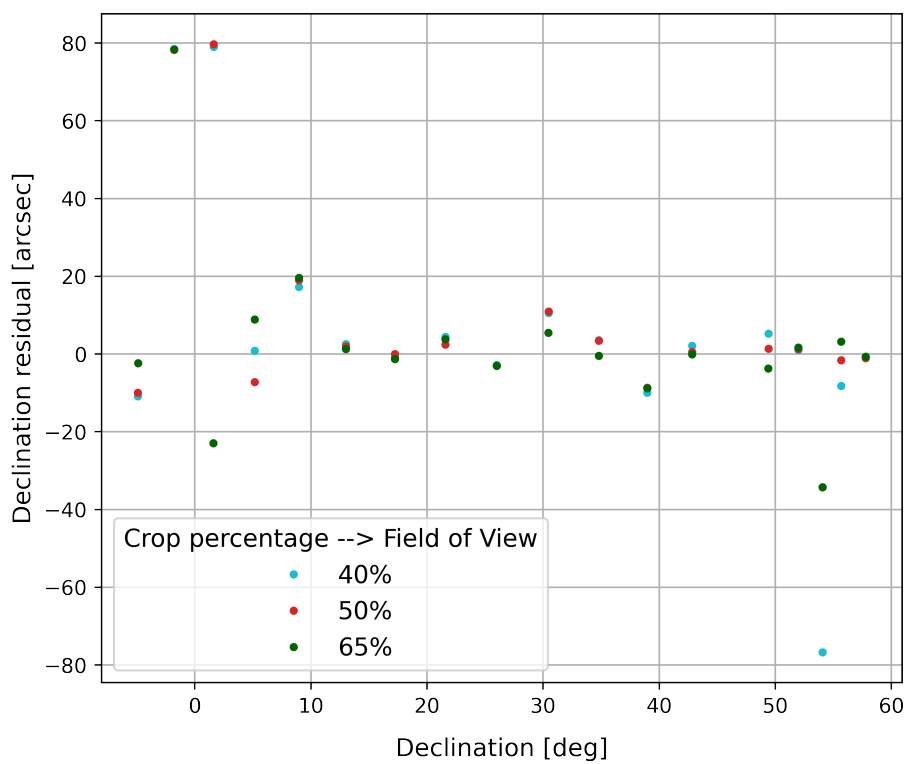
Accuracy in determining RADEC centroid coordinates

The first check on the accuracy in determining the position of the tracklet centroid and its RADEC coordinates is made by comparing the coordinates obtained from the cropped FITS images with those obtained through the analysis of the full-scale FITS images, which are taken as reference. Also in this case, only a subset of crop factors are considered, in particular those corresponding to the most optimal points in Fig. 4.12. The results obtained are shown in Fig. 4.13, referring to images coming from acquisition campaign number 2761, the same of Fig. 4.2 and Fig. 4.8. The results obtained from images of the other data acquisition campaigns are shown in the App. B (Fig. B.17 to Fig. B.20).

Also in this case, despite there is no loss of resolution, differences are observed on the residuals of the RADEC coordinates, even if slightly lower than in the previous cases in which the comparison took place with respect to images converted to PNG format.



(a)



(b)

Figure 4.13: Centroid RADEC residuals (cropped FITS images) - acquisition campaign number 2761

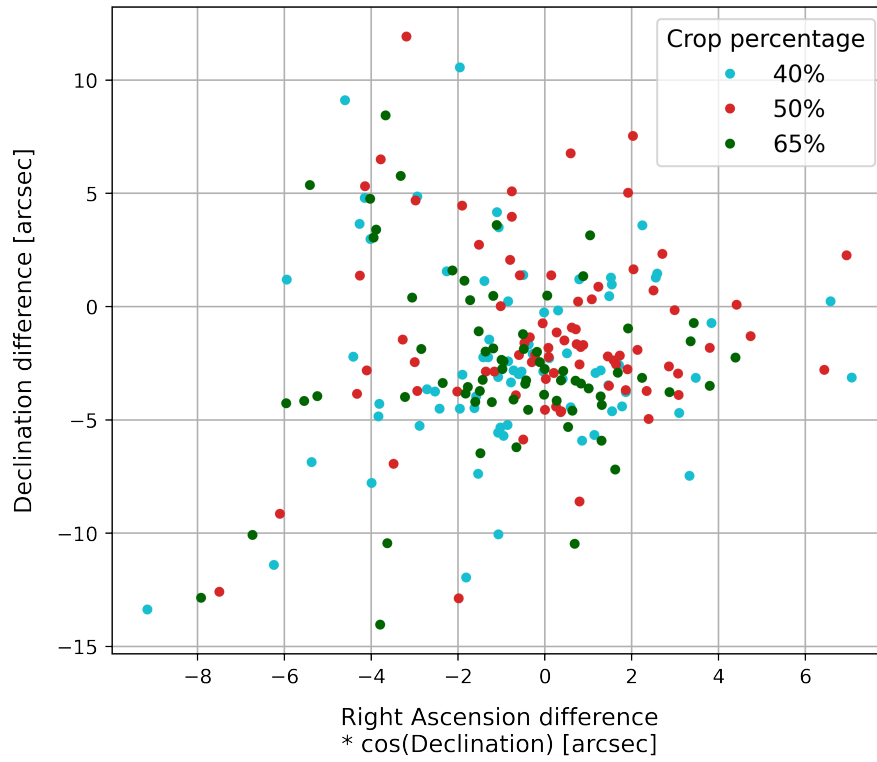
To check if the astrometric calibration process may be the cause of these differences in the centroid coordinates, a comparison is made between the pointing obtained from the center of the cropped FITS images and that obtained from the corresponding point, within the full-scale FITS image. The center of most of the cropped images coincides with the center of the bounding box that encloses the tracklet, barring inaccuracies introduced during the cropping step; the pixel coordinates of this point are known from the output of the tracklet detection process carried out with the RID tool. Knowing from the astrometric calibration process also the pointing of the center of the full-scale FITS image, it is possible to obtain the RADEC coordinates of the point coinciding with the center of the bounding box by applying an angular shift equal to the distance in pixels of this point with respect to the center of the full-scale image, multiplied by the *deg/pixel* ratio. The steps taken are reported in Alg. 4.1.

Algorithm 4.1 Pointing check

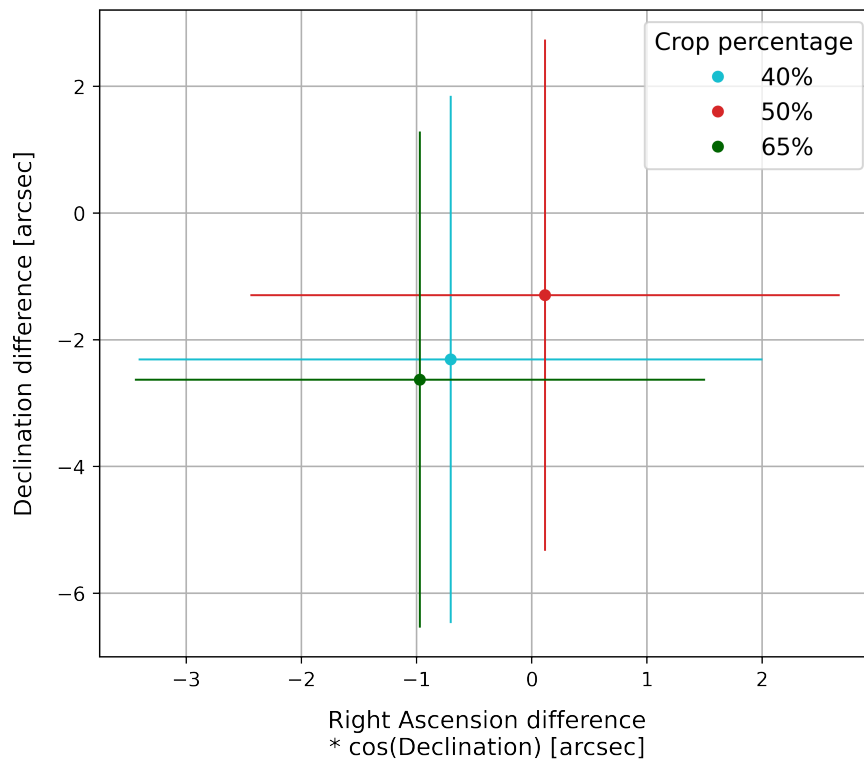
- 1: *Astrometric calibration* $\rightarrow (RA_{center}, DEC_{center})_{crop}$ % Pointing of cropped FITS
 - 2: *Astrometric calibration* $\rightarrow (RA_{center}, DEC_{center})_{full}$ % Pointing of full-scale FITS
 - 3: *Astrometric calibration* $\rightarrow deg/pix$ ratio % Value of *deg/pixel* ratio
 - 4: *Tracklet detection* $\rightarrow (c_x, c_y)_{BB}$ % Pixel coordinates of Bounding Box center
 - 5: $(O_x, O_y)_{full}$ % Known pixel coordinates of the center point of full-scale FITS
 - 6: $\Delta RA = (c_x - O_x) * (deg/pix \text{ ratio}) / \cos((DEC_{center})_{full})$ % Shift in RA
 - 7: $\Delta DEC = (c_y - O_y) * (deg/pix \text{ ratio})$ % Shift in DEC
 - 8: %% Comparison between pointing from cropped FITS image and reference from full-scale FITS image
 - 9: $Pointing_{crop} = (RA_{center}, DEC_{center})_{crop}$
 - 10: $Reference \ Pointing_{full} = (RA_{center}, DEC_{center})_{full} + (\Delta RA, \Delta DEC)$
-

The difference in pointing of the Right Ascension is computed taking into account the corrective factor $\cos DEC$. The results of this check are shown in Fig. 4.14. These values are probably overestimated due to misalignments in the order of magnitude of the *deg/pixel* ratio between the center of the image and the center of the bounding box, introduced both during the image cropping step and during the steps reported in Alg. 4.1.

A direct comparison between the difference in pointing and the difference in RADEC coordinates of the tracklet centroid is shown in Fig. 4.15, referring to the same subset of images used in the previous graphs. The RADEC centroid coordinates and pointing differences are compared between the full-scale and the cropped FITS (a crop factor equal to 0.5 is considered).

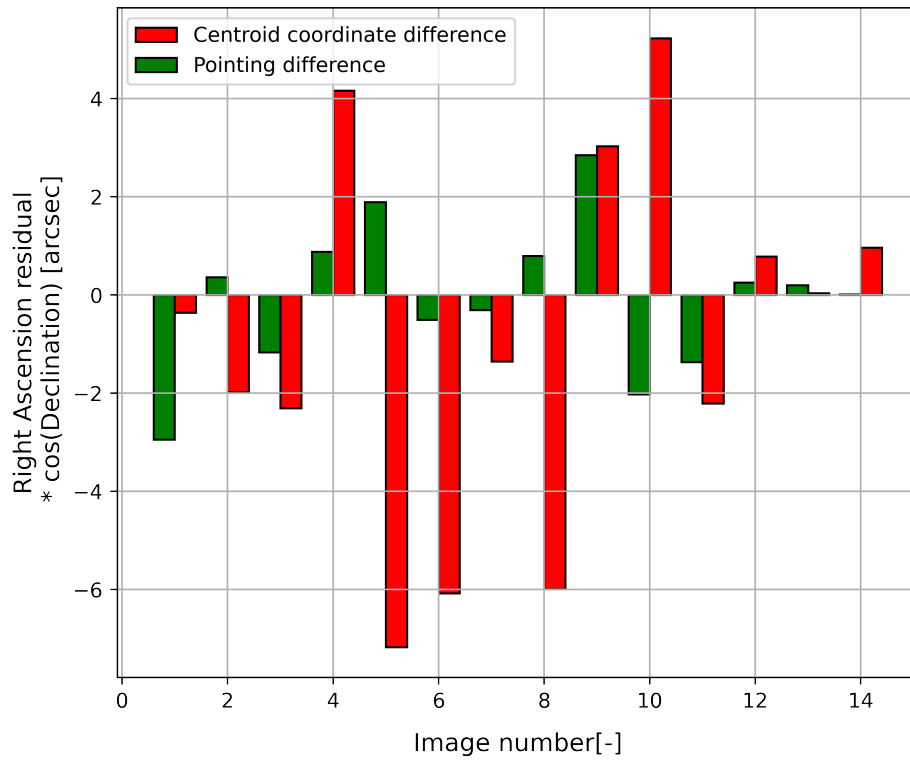


(a)

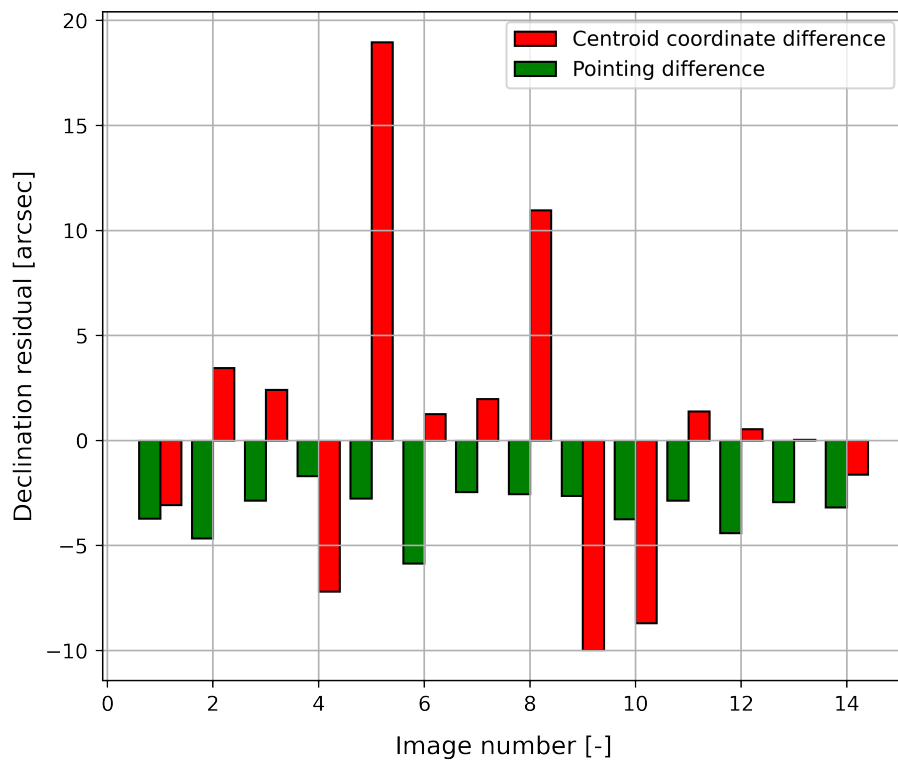


(b)

Figure 4.14: Pointing difference of cropped FITS images - All images (a) - Mean values (b)



(a)



(b)

Figure 4.15: Comparison of pointing and centroid coordinates differences between full-scale and cropped FITS images (campaign 2761)

The images characterized by a very high value of the centroid coordinates difference, attributable to a misidentification of the tracklet in the full-scale image or in the cropped image, were excluded from the graph shown in Fig. 4.15. Observing this graph, it is noted that the differences in pointing remain within a certain range, in the order of magnitude of the $deg/pixel$ ratio, while the differences in the centroid coordinates reach higher values. The same trend is observed also in the corresponding graphs, reported in App. B (Fig. B.21 to Fig. B.24). This is explained by a slightly different segmentation of the tracklet's contour points, comparing the cropped image to the full-scale image. This different identification of the edges, albeit slight, leads to a different value of the pixel coordinates of the centroid and, consequently, to different RADEC coordinates. Some images, both full-scale and cropped, were checked individually to verify the consistency between the observed difference in the pixel coordinates and the observed difference in the RADEC coordinates of the centroid, obtaining a positive feedback.

Finally, the check on RADEC centroid time-bias-corrected residuals calculated as the difference with respect to propagated TLE data is performed. Fig. 4.16 compares the time-bias-corrected residuals obtained from the original full-scale FITS images with those obtained from the cropped FITS images. The same graphs, referring to the remaining images from the other four acquisition campaigns, are showed in App. B (Fig. B.25 to Fig. B.28)

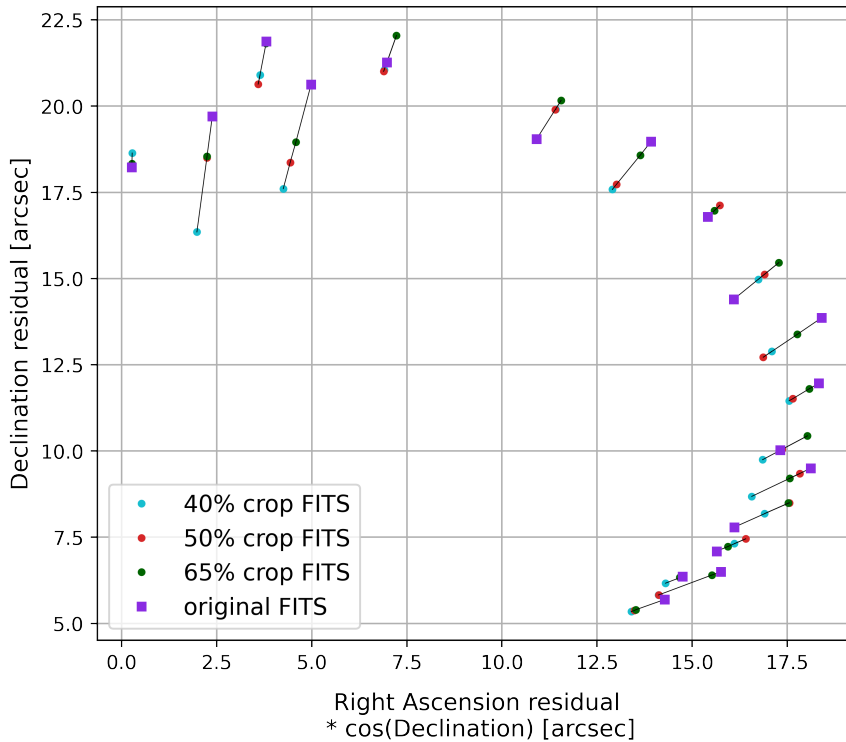


Figure 4.16: Centroid RADEC residuals corrected with time bias (campaign 2761)

Compared to the analogous graphs shown in Fig. 4.6 and Fig. 4.11 there is a smaller distance between the points representing the cropped images and the points representing the full-scale images and, above all, a trend that indicates greater accuracy associated with the full-scale images is not as evident.

Table 4.3 shows the average values of the RADEC centroid residuals corrected with the time-bias data for each data acquisition campaign, according to the different types of images analyzed.

Campaign	Coordinate	Full-scale FITS	50% crop FITS	50% crop PNG (Skimage)	50% crop PNG (Matplotlib)
2756	<i>RA cos DEC</i>	8.6834	8.9686	9.4415	9.1413
	<i>DEC</i>	14.9821	15.1835	16.3578	18.0213
2761	<i>RA cos DEC</i>	12.4134	12.2465	13.2382	13.7235
	<i>DEC</i>	13.8708	13.5439	14.5204	13.9952
2762	<i>RA cos DEC</i>	0.3289	0.1728	0.7769	1.0070
	<i>DEC</i>	2.0445	1.0295	4.7109	7.5317
2763	<i>RA cos DEC</i>	4.2885	3.7827	5.7966	6.5292
	<i>DEC</i>	4.9578	4.4663	7.4551	7.5883
2764	<i>RA cos DEC</i>	9.8006	9.9425	10.9071	12.9254
	<i>DEC</i>	11.8928	10.6264	14.4846	15.7634

Table 4.3: Mean values of RADEC centroid residuals corrected with time bias [arcsec]

Observing these data, it can be stated that the application of a crop around the original image in FITS format maintains substantially unchanged the accuracy with which the residuals of the RADEC coordinates corrected with the time-bias are obtained, starting from the location of the tracklet centroid. On the other hand, a decrease in accuracy is observed in cases where the analyzed images, before being cropped around the tracklet, undergo a process of down-scaling in resolution and down-conversion in pixel color depth.

5 | Conclusions and future developments

The approach proposed within this thesis is the result of the integration of an already existing pipeline for tracklet extraction and astrometric reduction with a Machine Learning aided object detection tool trained to detect the light streaks within images obtained from optical telescope acquisition and to indicate their position expressed in the image reference system. The upstream introduction of this tool before the original pipeline allows to obtain information about the presence of the tracklet in the image and its position. This allows to filter those images that do not contain any tracklet. If the tracklet is instead present, it is possible to define a portion of interest in the original image, containing the detected streak and the background starry sky. The original image can be cropped and then it can undergo the process of tracklet extraction.

Since the ML algorithm requires some image processing, resolution down-scaling, and image conversion into a new format, its introduction into the pipeline adds the possibility to choose which version of an image to analyze, the original one as it is obtained by the telescope, or the one processed, down-scaled in resolution, and converted to the new format. Two different image processing techniques were explored, each one associated with a different resolution down-scaling intensity.

The performances of the new pipeline were compared with those obtained from the original one. They were evaluated through three parameters, which are the pipeline success rate, intended as the percentage of images on which the tracklet extraction process is completed, the cost in terms of computational time taken to complete it, and the accuracy with which the angular coordinates of the framed object are obtained, after extracting the tracklet and carrying out the refinement process that allows having more regular tracklet shape.

Considering the optimal values of the crop factor and analyzing images in the converted format with lower resolution, relevant reductions in the computational time were obtained, while maintaining high success rates of the pipeline. The price to pay was a reduction in the accuracy with which the position of the tracklet centroid and therefore the angular

coordinates of the framed object were obtained. This is due to the loss of resolution, inherent in the transition to the new image format. In particular, the drop in computational time went from the 35 seconds per image necessary for the analysis of the original image, to the 12 or 6.5 seconds per image needed for the analysis of the cropped, resolution down-scaled image, depending on the chosen image processing technique.

Maintaining the original image format and resolution and considering optimal crop values, a reduction in the computational cost was observed, even if less marked with respect to the previous case, passing from 35 to 24 seconds per image. A high success rate of the analysis was maintained and no significant variations in the accuracy with which the position of the tracklet's centroid is obtained were noted, even if the latter was observed to be almost always slightly different from that obtained by analyzing the original image. This led to the conclusion that the new pipeline architecture is advantageous concerning the first two parameters of merit considered for the comparison, but that it is not enough for improving the precision in obtaining the tracklet centroid. It is therefore believed that the "bottleneck" lies in the tracklet segmentation process through the edge-detection algorithm. This process is ruled by a set of parameters, the values of which determine how the segmentation of the tracklet takes place. Fine-tuning these parameters would be necessary to obtain an optimal detection of the contour points of the tracklet for each image but this would require continuous manual intervention.

A possible way to make less variable the result on the position of the tracklet centroid could be to apply a more "aggressive" refinement, which allows to obtain a smoother tracklet contour. In addition, once the edges of the tracklet have been determined, it could be advantageous to obtain a binary mask to distinguish the internal points of the tracklet from the image background, and compute the centroid position as the average of their coordinates. This could further limit the noisy effect observed in the tracklet's contour.

Bibliography

- [1] European Space Agency. Space debris by the numbers. https://www.esa.int/Space_Safety/Space_Debris/Space_debris_by_the_numbers, August 2022. Accessed on 01-09-2022.
- [2] ESA Space Debris Office. ESA'S ANNUAL SPACE ENVIRONMENT REPORT. https://www.sdo.esoc.esa.int/environment_report/Space_Environment_Report_latest.pdf, 2022. Accessed on 01-09-2022.
- [3] European Space Agency. About space debris. https://www.esa.int/Space_Safety/Space_Debris/About_space_debris. Accessed on 01-09-2022.
- [4] Thomas G. Roberts. Popular orbits 101. <https://aerospace.csis.org/aerospace101/popular-orbits-101/>. Accessed on 01-09-2022.
- [5] D. Leonard. Space Junk Removal Is Not Going Smoothly. <https://www.scientificamerican.com/article/space-junk-removal-is-not-going-smoothly/>, April 2021. Accessed on 01-09-2022.
- [6] Donald J Kessler and Burton G Cour-Palais. Collision frequency of artificial satellites: The creation of a debris belt. *Journal of Geophysical Research: Space Physics*, 83(A6):2637–2646, 1978.
- [7] European Space Agency. SSA Programme overview. https://www.esa.int/Space_Safety/SSA_Programme_overview. Accessed on 01-09-2022.
- [8] European Space Agency. Space Surveillance and Tracking - SST Segment. https://www.esa.int/Space_Safety/Space_Surveillance_and_Tracking_-_SST_Segment. Accessed on 01-09-2022.
- [9] Alessandro Morselli. High order methods for Space Situational Awareness. Master's thesis, Politecnico di Milano, School of industrial and information engineering, Department of aerospace science and technology, Italy, 2014.

- [10] Junju Du, Shaoming Hu, Xu Chen, and Difu Guo. Improved space debris astrometry with template matching. 2019.
- [11] G.Lavezzi. Image processing of multiclass satellite tracklets for initial orbit determination based on optical telescopes. Master's thesis, Politecnico di Milano, School of industrial and information engineering, Department of aerospace science and technology, Italy, 2018.
- [12] J.Africano et al. Understanding photometric phase angle corrections. In *4th European Conference on Space Debris*, volume 587 of *ESA Special Publication*, page 141. Danesy, Aug.2005.
- [13] Riccardo Cipollone and Andrea De Vittori. Machine Learning Techniques for Optical and Multibeam Radar Track Reconstruction of LEO Objects. Master's thesis, Politecnico di Milano, School of industrial and information engineering, Department of aerospace science and technology, Italy, 2020.
- [14] Dae-Won Kim. ASTRiDE: Automated Streak Detection for Astronomical Images. <https://github.com/dwkim78/ASTRiDE>, May 2016. Accessed on 01-09-2022.
- [15] Vladimir Koupryanov. Distinguishing features of ccd astrometry of faint geo objects. *Advances in Space Research*, 41(7):1029–1038, 2008.
- [16] Stephanie Gibson et al. *Comparison of PSF fitting methods for determining centroids of stars*. PhD thesis, Massachusetts Institute of Technology, 2013.
- [17] Roberto Brunelli. *Template matching techniques in computer vision: theory and practice*. John Wiley & Sons, 2009.
- [18] Navigation and Ancillary Information Facility. An Overview of Reference Frames and Coordinate Systems in the SPICE Context. https://naif.jpl.nasa.gov/pub/naif/toolkit_docs/Tutorials/pdf/individual_docs/17_frames_and_coordinate_systems.pdf, January 2020. Accessed on 01-09-2022.
- [19] H. Karttunen, P. Kröger, H. Oja, M. Poutanen, and K. J. Donner. *Fundamental Astronomy*, chapter 3, pages 47–54, 69–71. Springer, 5 edition, 2007.
- [20] TELEDYNE DALSA. Ccd vs. cmos. <https://www.teledynedalsa.com/en/learn/knowledge-center/ccd-vs-cmos/>. Accessed on 01-09-2022.
- [21] Mauro Massari. Optical instruments ppt presentation. Politecnico di Milano, School

- of industrial and information engineering, Department of aerospace science and technology, March 2020.
- [22] Justin T Cooper, Ines Juvan-Beaulieu, and Adam J. Wise. Scientific cmos optical detectors for orbital debris observations. 2019.
- [23] Aperture, focal length and focal ratio. <https://blog.optics-trade.eu/aperture/>, 2019. Accessed on 01-09-2022.
- [24] Giovanni M Del Genio, Jacopo Paoli, Enrico Del Grande, Ferdinando Dolce, Walter Villadei, Marco Reali, and Andrea Aquilini. Italian air force radar and optical sensor experiments for the detection of space objects in leo orbit. In *Proceedings of the 16th Advanced Maui Optical and Space Surveillance Technologies Conference*, 2015.
- [25] Serg. A. Mangiacapre. Forniti i primi servizi di Space Surveillance & Tracking. https://www.aeronautica.difesa.it/missione/versofuturo/GS17/news_GS17/Pagine/primi_servizi_SpaceSurveillance_Tracking.aspx, April 2017. Accessed on 01-09-2022.
- [26] Flexible Image Transport System (FITS), Version 3.0. <https://www.loc.gov/preservation/digital/formats/fdd/fdd000317.shtml#notes>. Accessed on 01-09-2022.
- [27] Jason Calvi. Machine Learning Techniques for Detection and Tracking of Space Objects in Optical Telescope Images. Master’s thesis, Politecnico di Milano, School of industrial and information engineering, Department of aerospace science and technology, Italy, 2021.
- [28] FITS World Coordinate System (WCS). https://fits.gsfc.nasa.gov/fits_wcs.html. Accessed on 01-09-2022.
- [29] Dustin Lang, David W Hogg, Keir Mierle, Michael Blanton, and Sam Roweis. Astrometry. net: Blind astrometric calibration of arbitrary astronomical images. *The astronomical journal*, 139(5):1782, 2010.
- [30] Joaquín González López-Cepero. AstroDART: Astronomical data analysis and recovery from tracklets. Master’s thesis, Politecnico di Milano, School of industrial and information engineering, Department of aerospace science and technology, Italy, 2022.

A | Appendix A

The tables in this appendix contain, respectively, the data from which the graphs in Fig. 4.1, Fig. 4.7, and Fig. 4.12 are obtained. The computational cost and the success rate associated with the most significant modules of the pipeline are reported, according to the variation in the dimensions of the images.

Image format	Crop factor [-]	Tracklet detection time [s/image]	Astrometric calibration time [s/image]	Astrometric calibration success [%]	Tracklet extraction time [s/image]	Tracklet extraction success [%]
PNG	0.20	5.62	11.85	68.67	3.26	100.00
PNG	0.25	5.62	4.91	89.89	3.73	100.00
PNG	0.30	5.62	1.91	98.88	4.14	98.86
PNG	0.35	5.62	1.53	98.88	4.38	97.73
PNG	0.40	5.62	1.66	98.88	4.40	100.00
PNG	0.45	5.62	1.47	98.88	4.22	100.00
PNG	0.50	5.62	1.50	100.00	4.34	98.88
PNG	0.55	5.62	1.65	100.00	4.49	100.00
PNG	0.60	5.62	1.88	100.00	4.72	100.00
PNG	0.65	5.62	2.05	100.00	4.84	97.75
PNG	0.70	5.62	3.21	100.00	5.46	98.86
PNG	0.75	5.62	4.95	100.00	5.73	98.86
PNG	0.80	5.62	7.49	98.88	5.84	100.00
PNG	0.85	5.62	8.55	98.88	5.82	98.85
PNG	0.90	5.62	9.97	98.88	6.76	97.70
PNG	0.95	5.62	13.94	100.00	7.55	96.59
PNG	1	5.62	14.29	98.88	7.78	94.25
FITS	1	-	11.50	100.00	23.94	100.00

Table A.1: Success rate and computational time analyzing images in PNG format with different crop size (Scikit-image processing)

Image format	Crop factor [-]	Tracklet detection time [s/image]	Astrometric calibration time [s/image]	Astrometric calibration success [%]	Tracklet extraction time [s/image]	Tracklet extraction success [%]
PNG	0.20	0.81	6.36	14.81	2.31	91.67
PNG	0.25	0.81	3.20	30.23	2.39	96.15
PNG	0.30	0.81	3.32	52.27	2.81	93.48
PNG	0.35	0.81	3.40	73.86	3.13	92.31
PNG	0.40	0.81	1.92	92.05	3.47	92.59
PNG	0.45	0.81	1.52	97.73	3.81	91.86
PNG	0.50	0.81	1.50	98.86	3.94	91.95
PNG	0.55	0.81	1.52	98.86	3.63	90.80
PNG	0.60	0.81	1.54	98.82	3.56	92.86
PNG	0.65	0.81	1.89	98.82	3.48	88.10
PNG	0.70	0.81	1.93	98.82	3.50	89.29
PNG	0.75	0.81	2.12	98.82	3.48	86.90
PNG	0.80	0.81	2.45	98.82	3.55	86.90
PNG	0.85	0.81	3.05	98.81	3.68	90.36
PNG	0.90	0.81	3.25	97.56	3.60	91.25
PNG	0.95	0.81	3.37	97.56	3.67	90.00
PNG	1	0.81	3.31	100.00	4.13	87.65
FITS	1	-	11.50	100.00	23.94	100.00

Table A.2: Success rate and computational time analyzing images in PNG format with different crop size (Matplotlib processing)

Image format	Crop factor [-]	Tracklet detection time [s/image]	Astrometric calibration time [s/image]	Astrometric calibration success [%]	Tracklet extraction time [s/image]	Tracklet extraction success [%]
FITS	0.20	5.76	2.67	27.38	4.12	95.65
FITS	0.25	5.76	2.85	56.18	6.14	96.00
FITS	0.30	5.76	2.67	82.02	7.09	97.26
FITS	0.35	5.76	2.50	95.51	11.30	97.65
FITS	0.40	5.76	2.69	96.55	10.00	96.43
FITS	0.45	5.76	3.37	98.85	12.07	97.67
FITS	0.50	5.76	4.18	98.85	13.97	98.84
FITS	0.55	5.76	4.33	98.85	15.29	96.51
FITS	0.60	5.76	4.96	98.85	19.24	95.35
FITS	0.65	5.76	5.42	98.85	15.98	100.00
FITS	0.70	5.76	5.98	98.85	16.95	98.84
FITS	0.75	5.76	7.34	98.85	21.84	96.51
FITS	0.80	5.76	7.47	100.00	24.63	95.40
FITS	0.85	5.76	8.56	100.00	23.92	97.70
FITS	0.90	5.76	10.74	98.85	24.30	96.51
FITS	0.95	5.76	10.62	98.85	26.37	97.67
FITS	1	-	11.50	100.00	23.94	100.00

Table A-3: Success rate and computational time analyzing images in FITS format with different crop size

B | Appendix B

This appendix shows the graphs obtained from the remaining data acquisition campaigns, corresponding to those shown in Chap. 4 relative to the subset of images obtained from campaign number 2761.

Scikit-image processing: accuracy in determining RADEC centroid coordinates

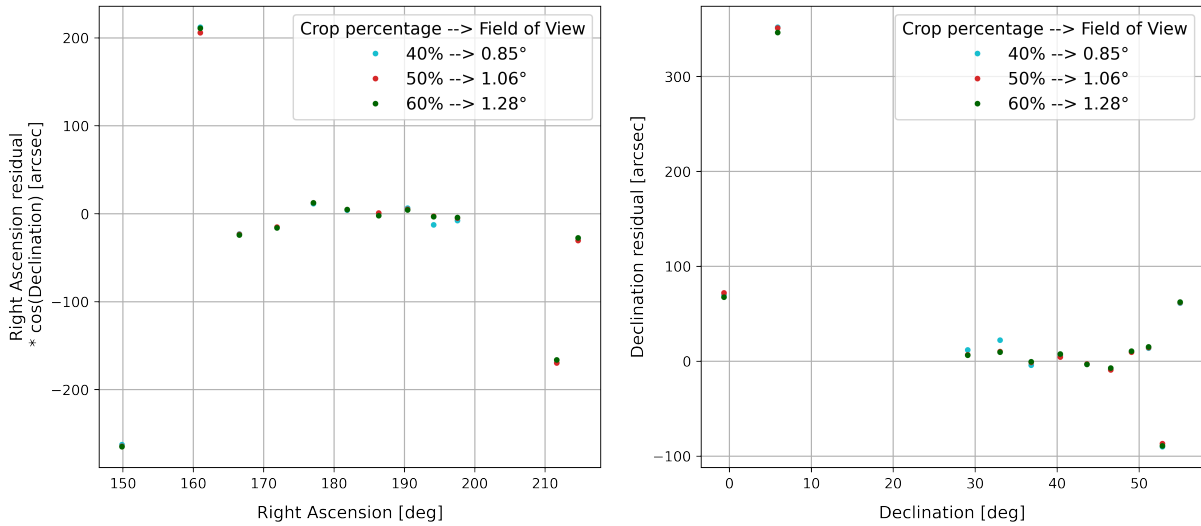


Figure B.1: Centroid RADEC residuals (Scikit-image processing) - acquisition campaign number 2756

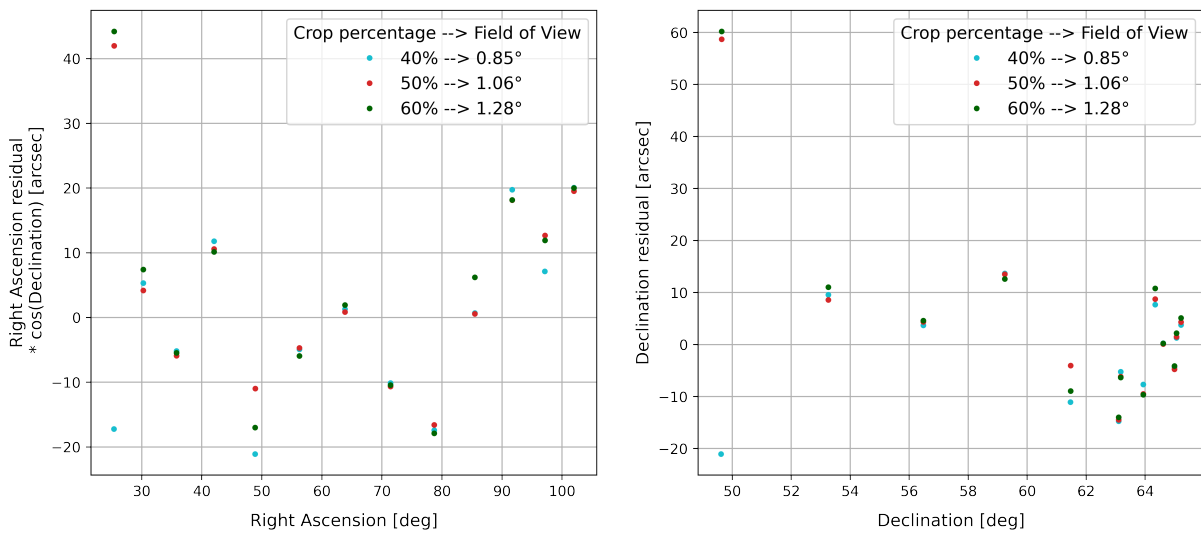


Figure B.2: Centroid RADEC residuals (Scikit-image processing) - acquisition campaign number 2762

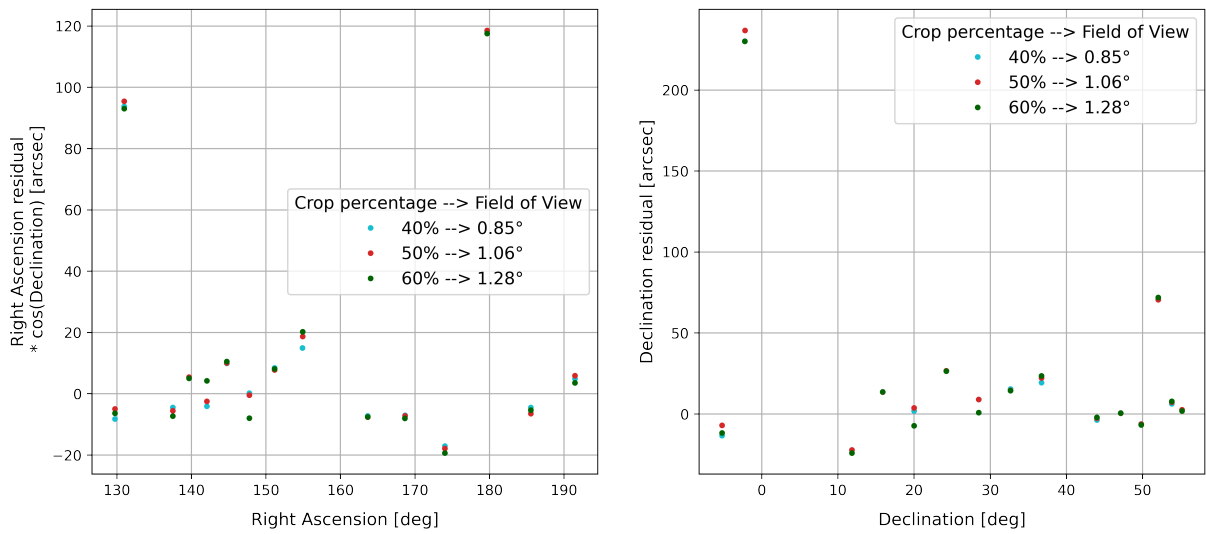


Figure B.3: Centroid RADEC residuals (Scikit-image processing) - acquisition campaign number 2763

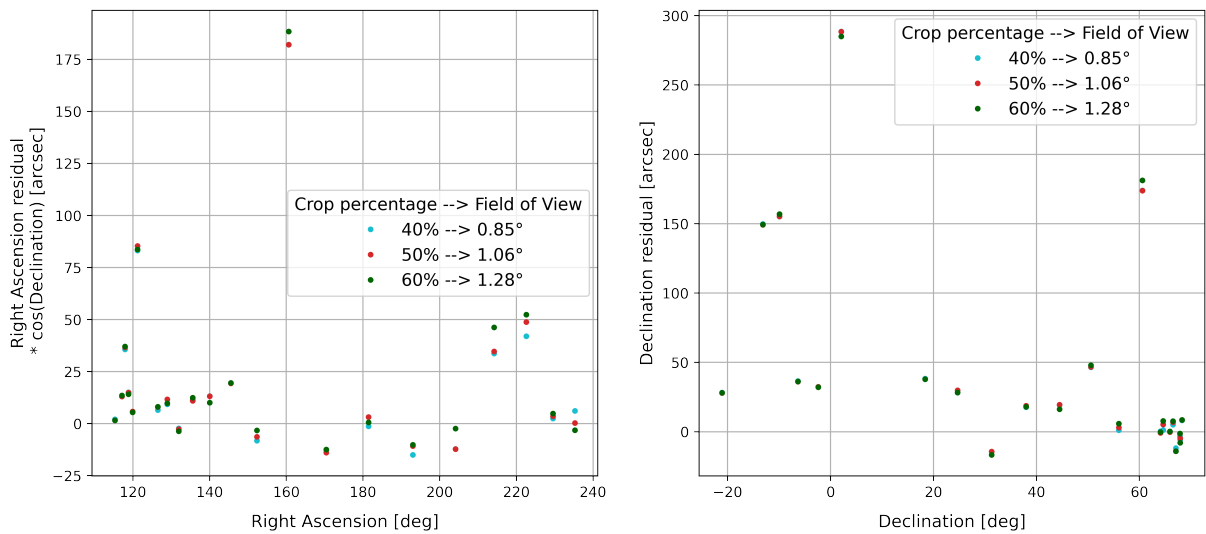


Figure B.4: Centroid RADEC residuals (Scikit-image processing) - acquisition campaign number 2764

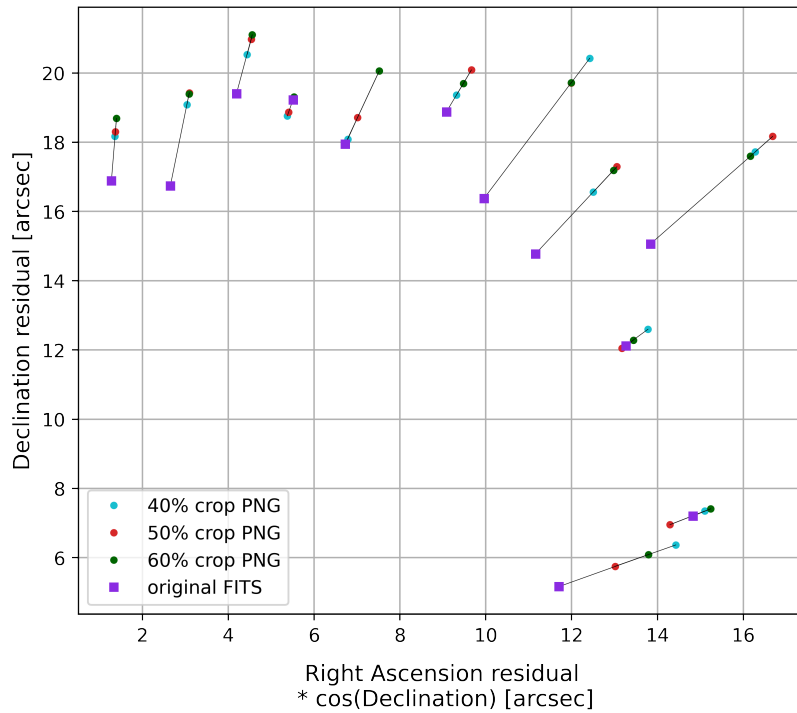


Figure B.5: Centroid RADEC residuals corrected with time bias (campaign 2756) - Scikit-image processing

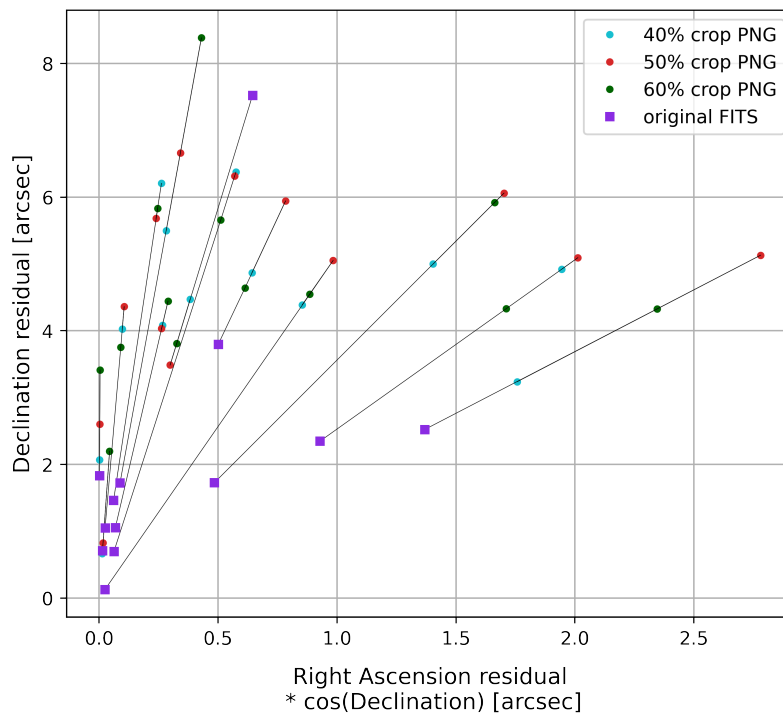


Figure B.6: Centroid RADEC residuals corrected with time bias (campaign 2762) - Scikit-image processing

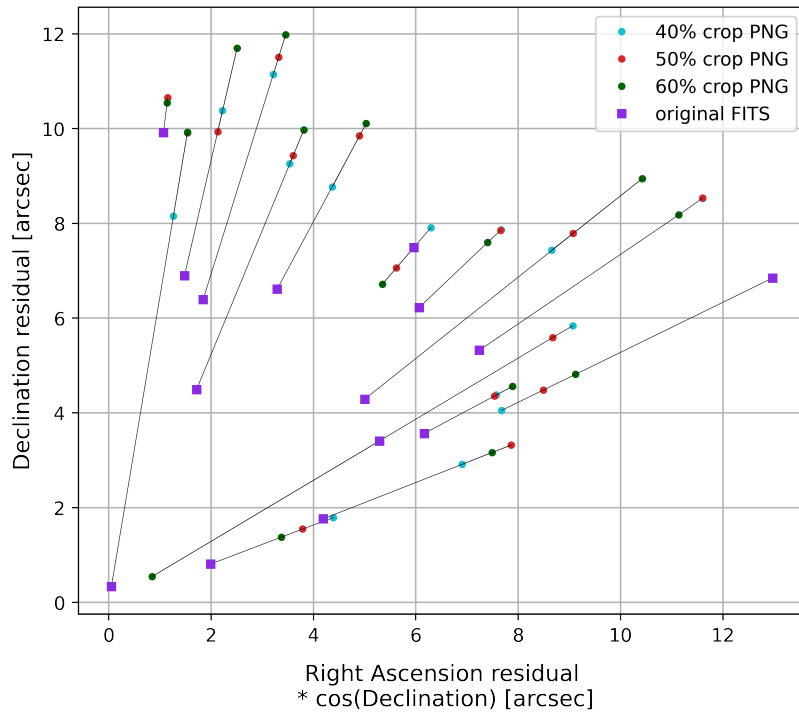


Figure B.7: Centroid RADEC residuals corrected with time bias (campaign 2763) - Scikit-image processing

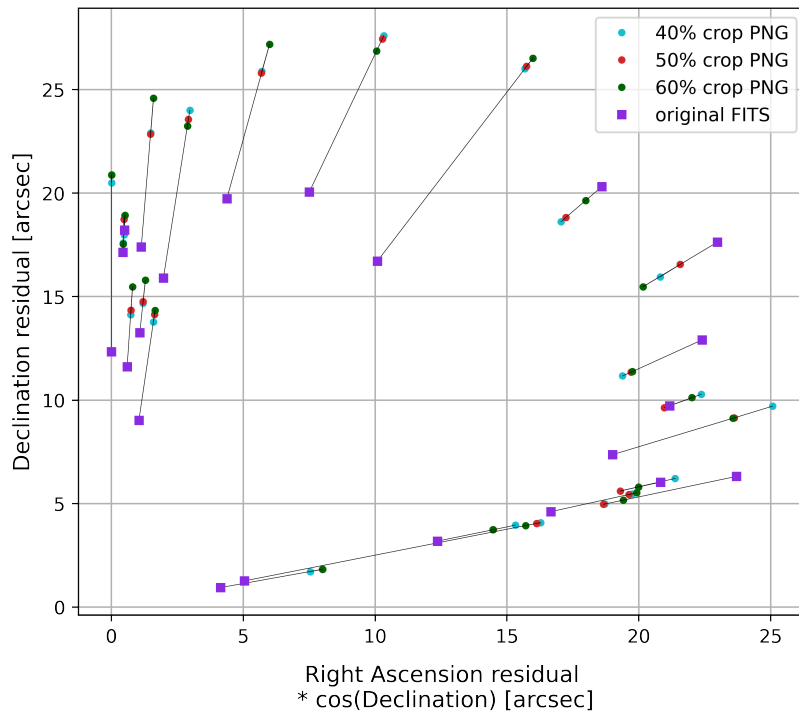


Figure B.8: Centroid RADEC residuals corrected with time bias (campaign 2764) - Scikit-image processing

Matplotlib processing: accuracy in determining RADEC centroid coordinates

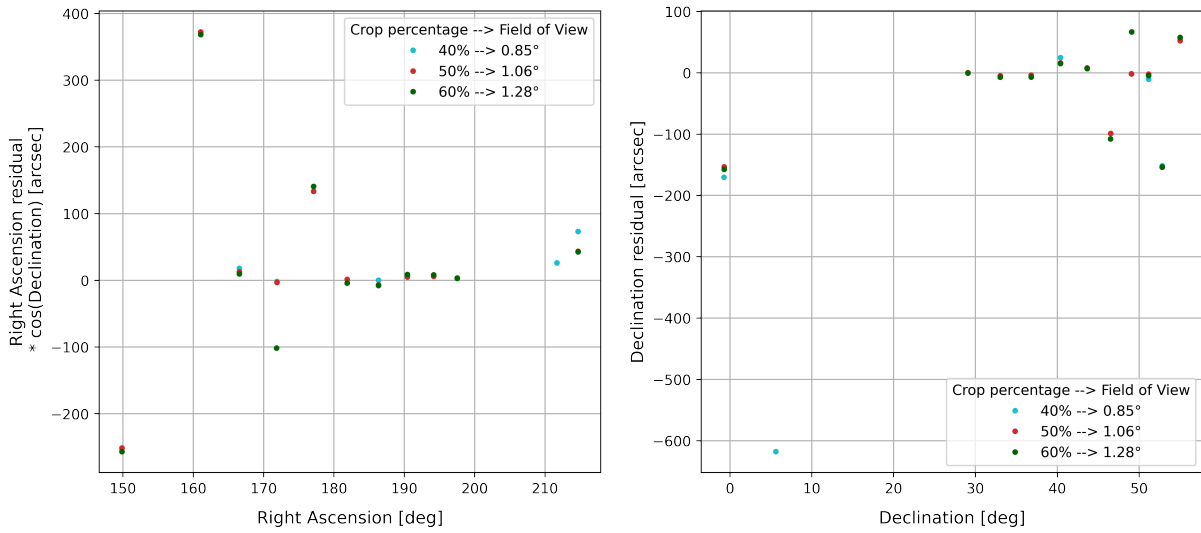


Figure B.9: Centroid RADEC residuals (Matplotlib processing) - acquisition campaign number 2756

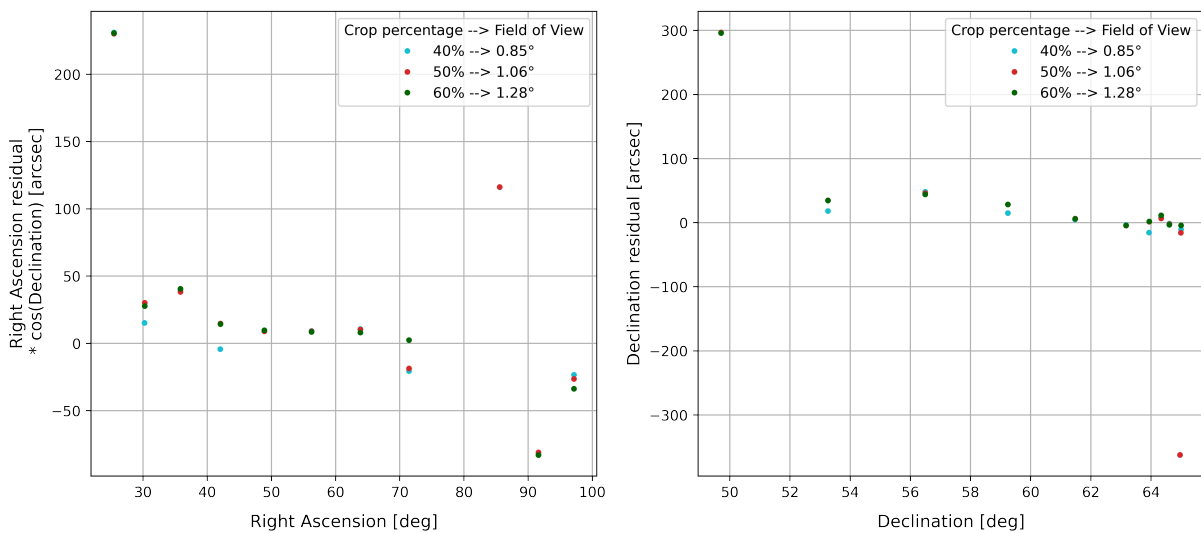


Figure B.10: Centroid RADEC residuals (Matplotlib processing) - acquisition campaign number 2762

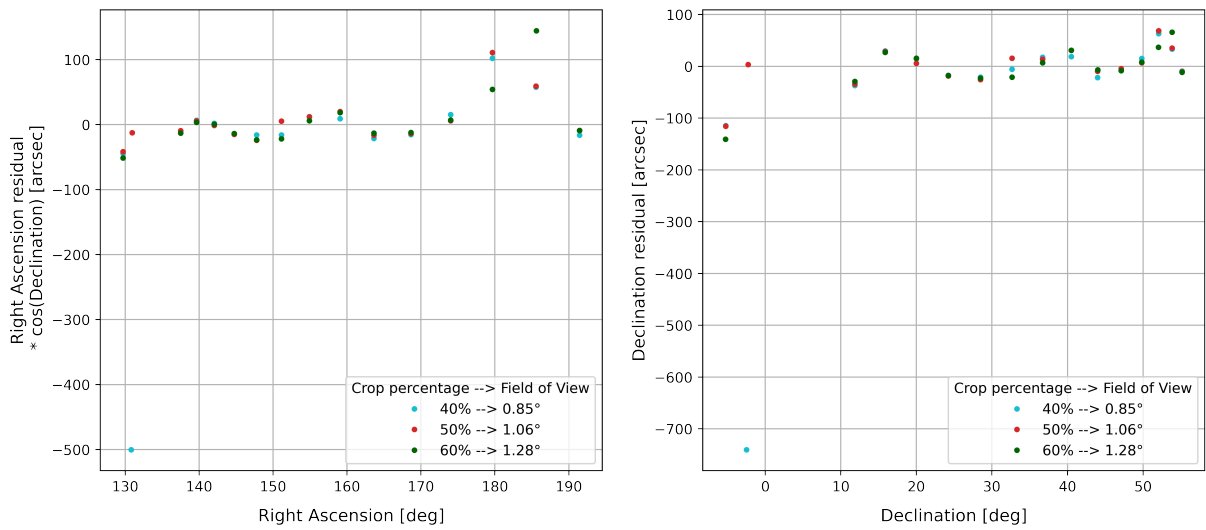


Figure B.11: Centroid RADEC residuals (Matplotlib processing) - acquisition campaign number 2763

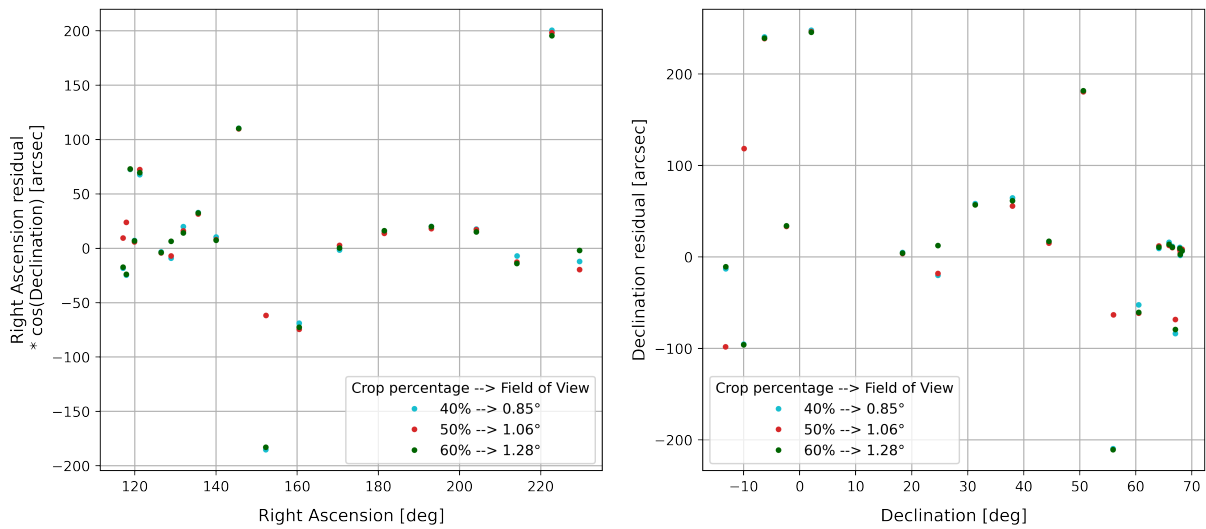


Figure B.12: Centroid RADEC residuals (Matplotlib processing) - acquisition campaign number 2764

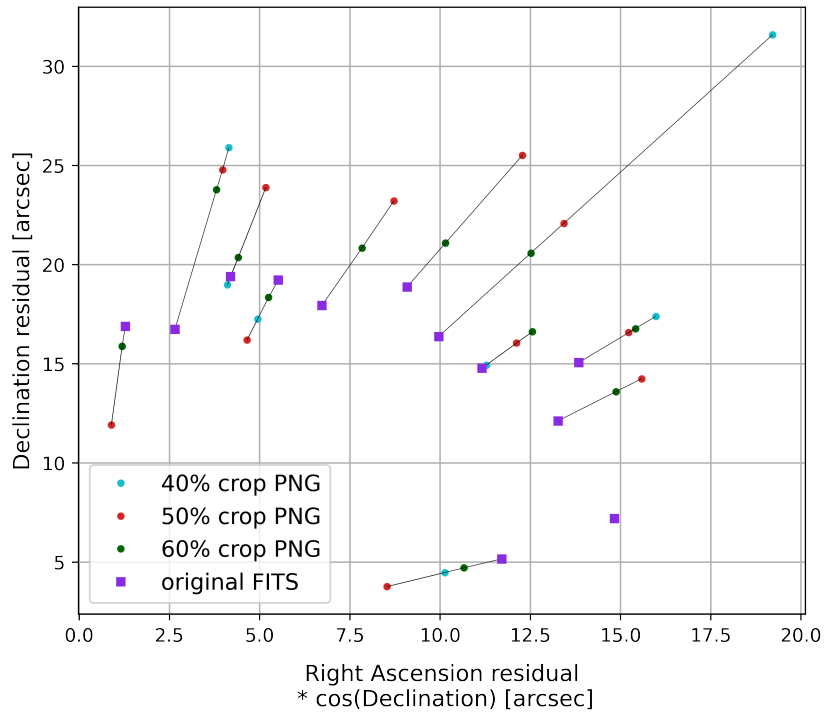


Figure B.13: Centroid RADEC residuals corrected with time bias (campaign 2756) - Matplotlib processing

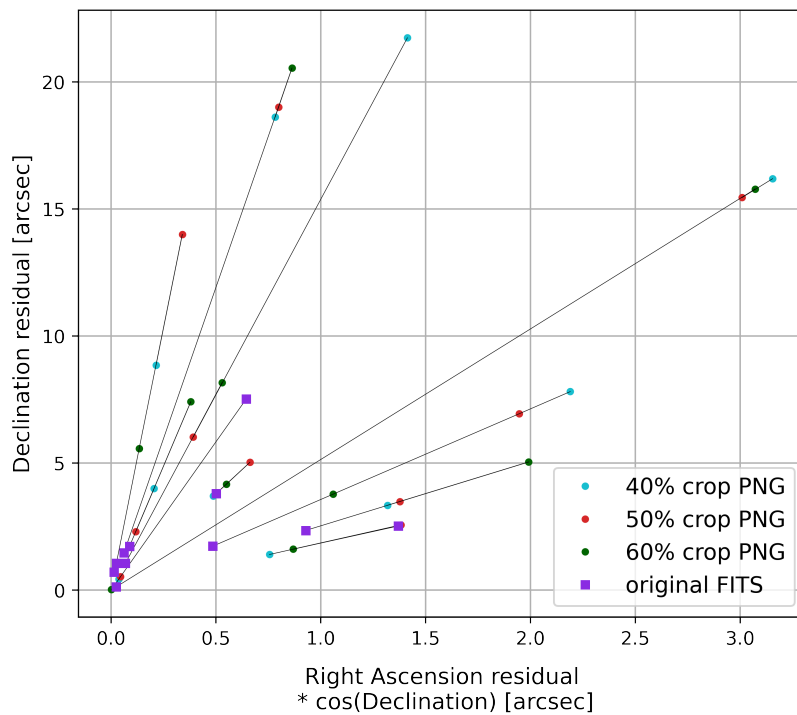


Figure B.14: Centroid RADEC residuals corrected with time bias (campaign 2762) - Matplotlib processing

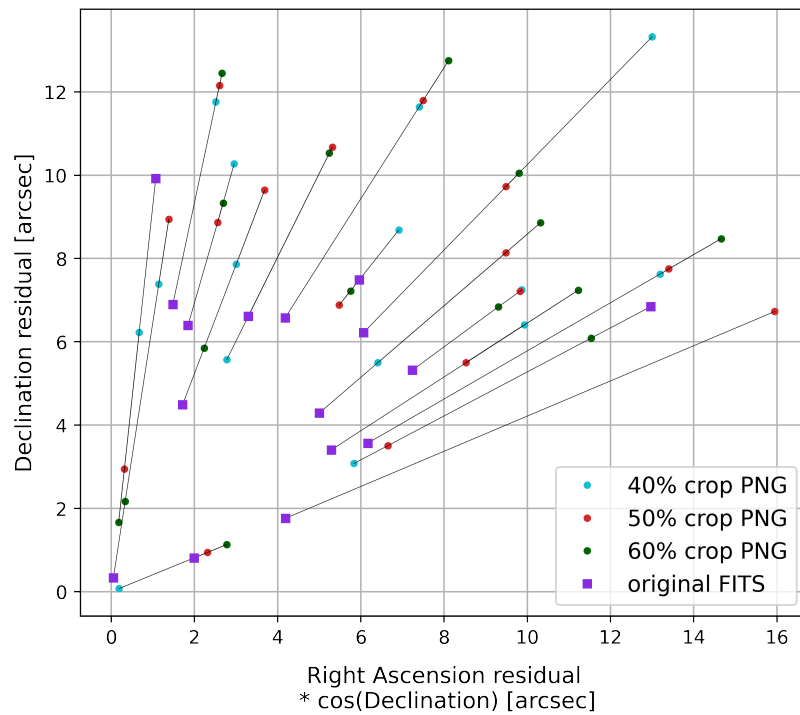


Figure B.15: Centroid RADEC residuals corrected with time bias (campaign 2763) - Matplotlib processing

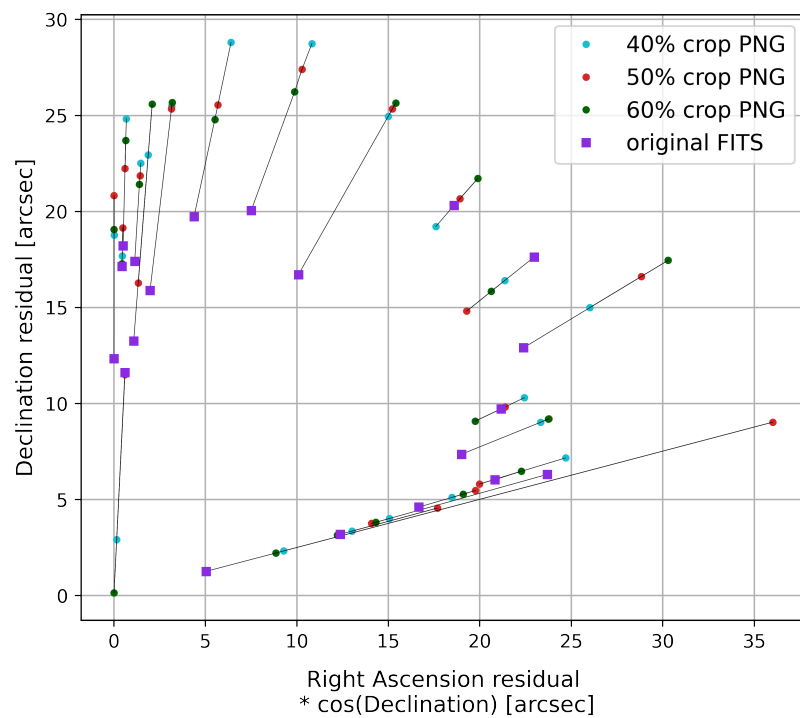


Figure B.16: Centroid RADEC residuals corrected with time bias (campaign 2764) - Matplotlib processing

Cropped FITS images: accuracy in determining RADEC centroid coordinates

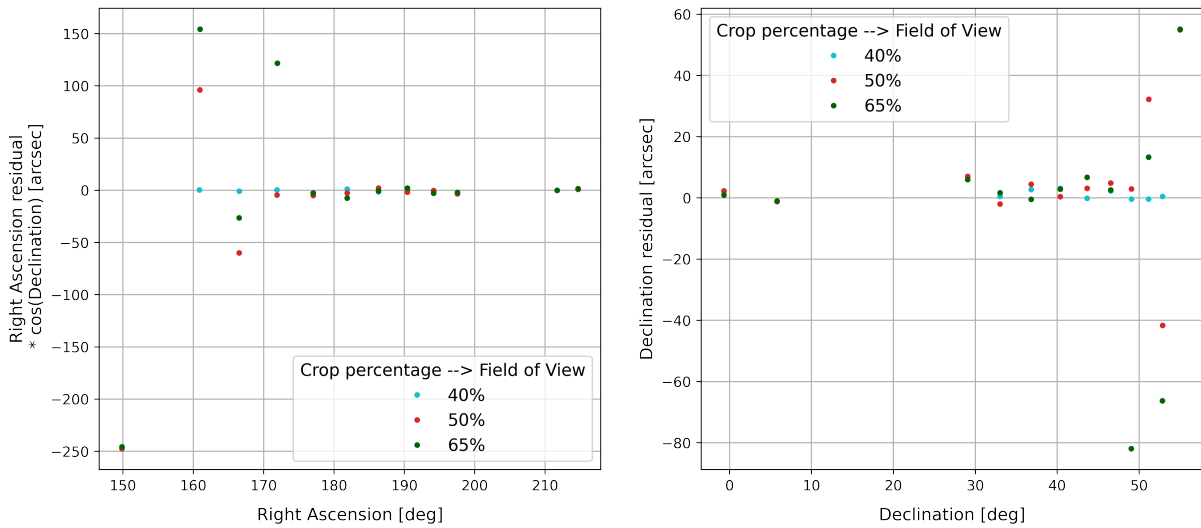


Figure B.17: Centroid RADEC residuals (cropped FITS images) - acquisition campaign number 2756

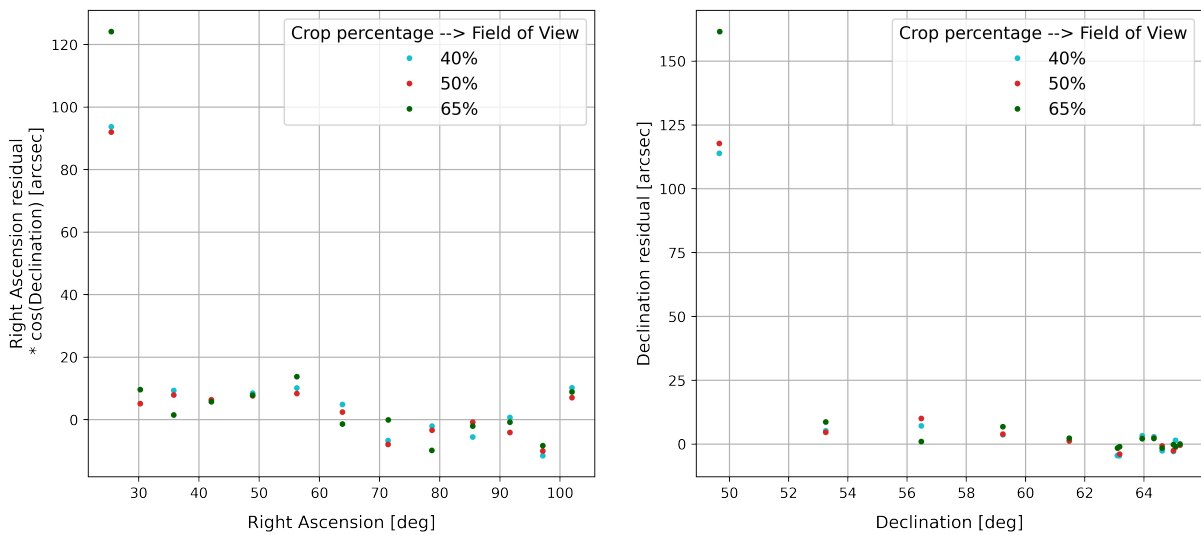


Figure B.18: Centroid RADEC residuals (cropped FITS images) - acquisition campaign number 2762

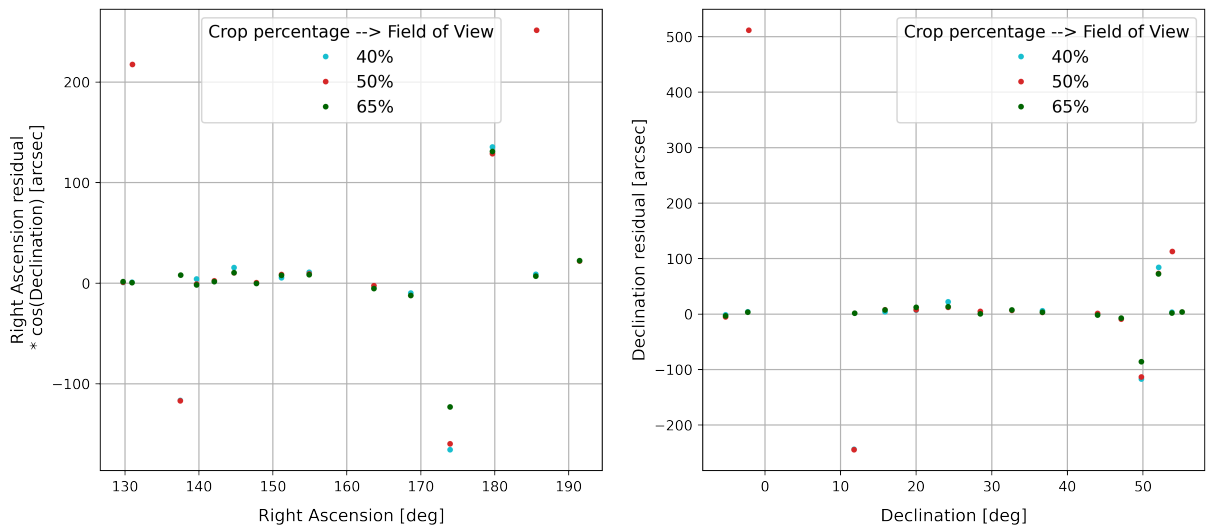


Figure B.19: Centroid RADEC residuals (cropped FITS images) - acquisition campaign number 2763

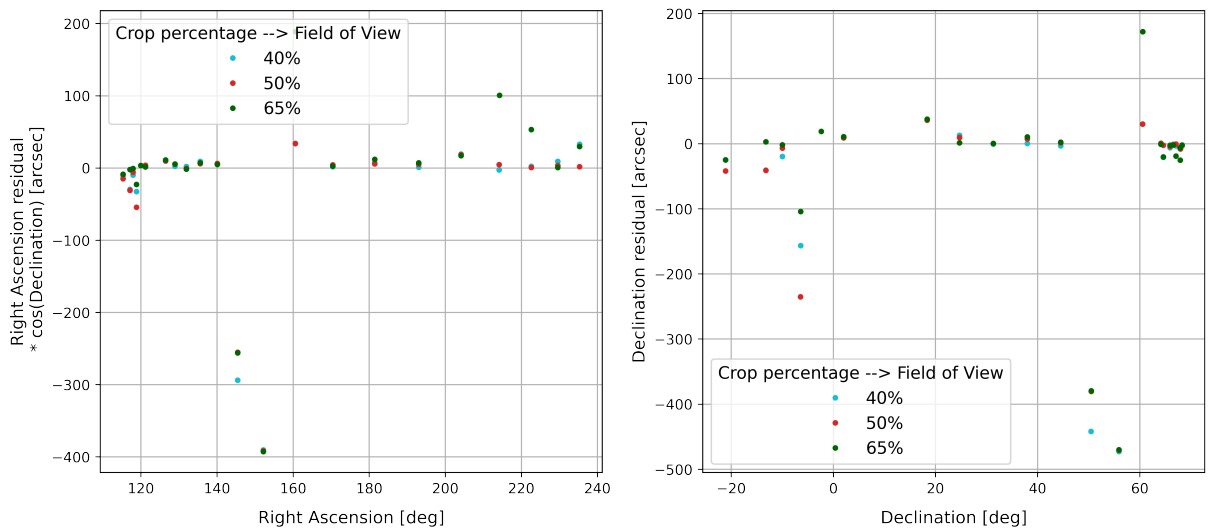


Figure B.20: Centroid RADEC residuals (cropped FITS images) - acquisition campaign number 2764

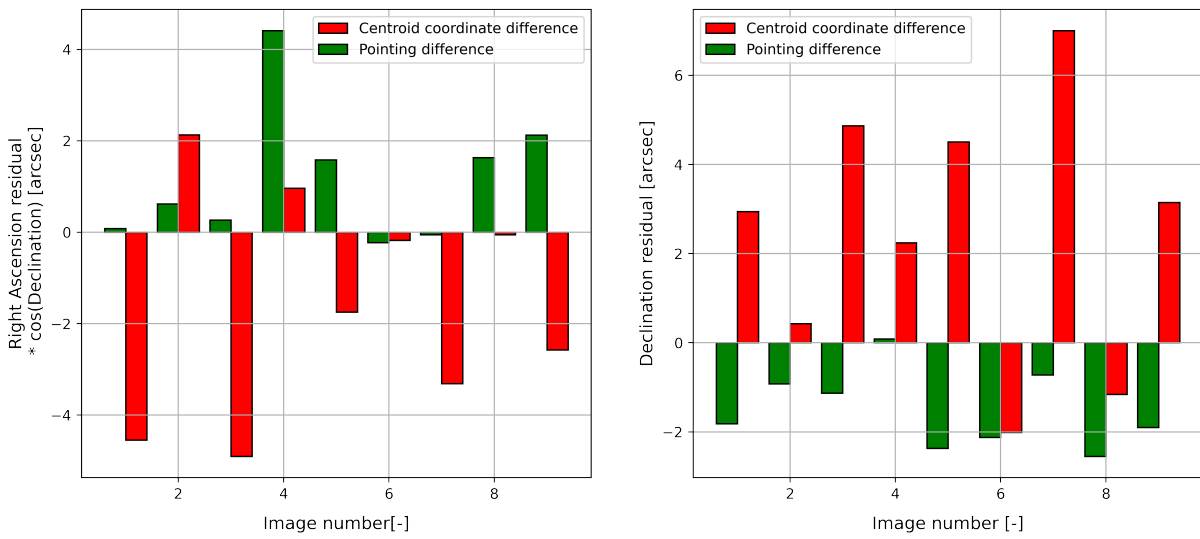


Figure B.21: Comparison of pointing and centroid coordinates differences between full-scale and cropped FITS images (campaign 2756)

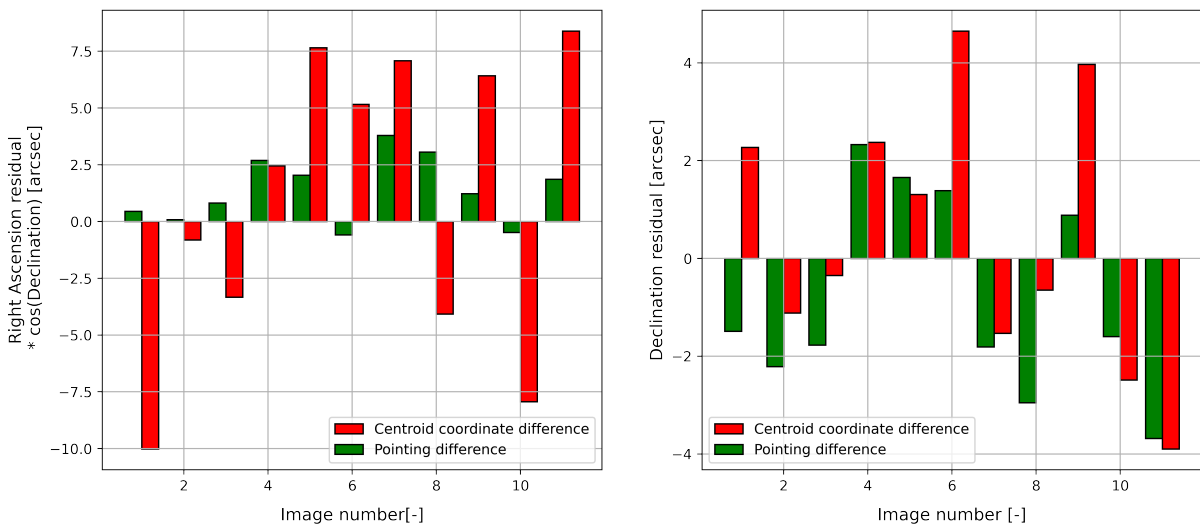


Figure B.22: Comparison of pointing and centroid coordinates differences between full-scale and cropped FITS images (campaign 2762)

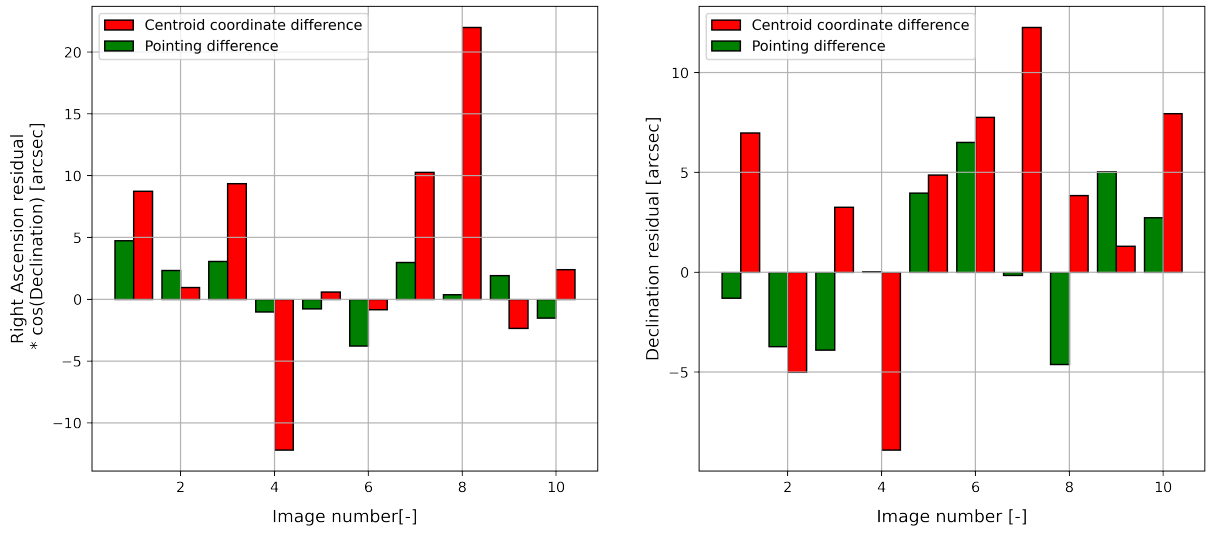


Figure B.23: Comparison of pointing and centroid coordinates differences between full-scale and cropped FITS images (campaign 2763)

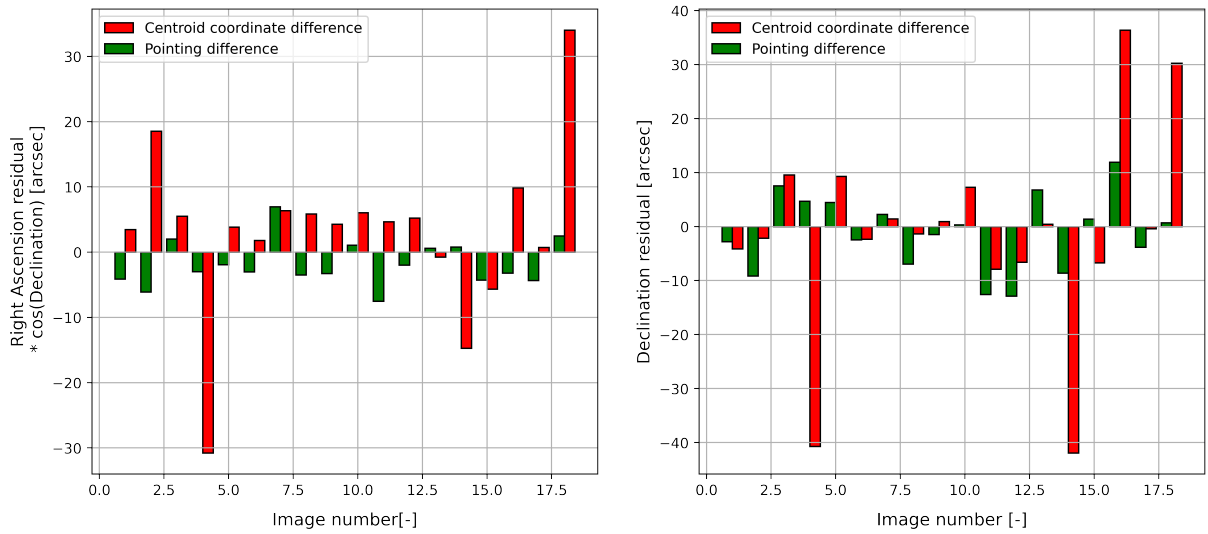


Figure B.24: Comparison of pointing and centroid coordinates differences between full-scale and cropped FITS images (campaign 2764)

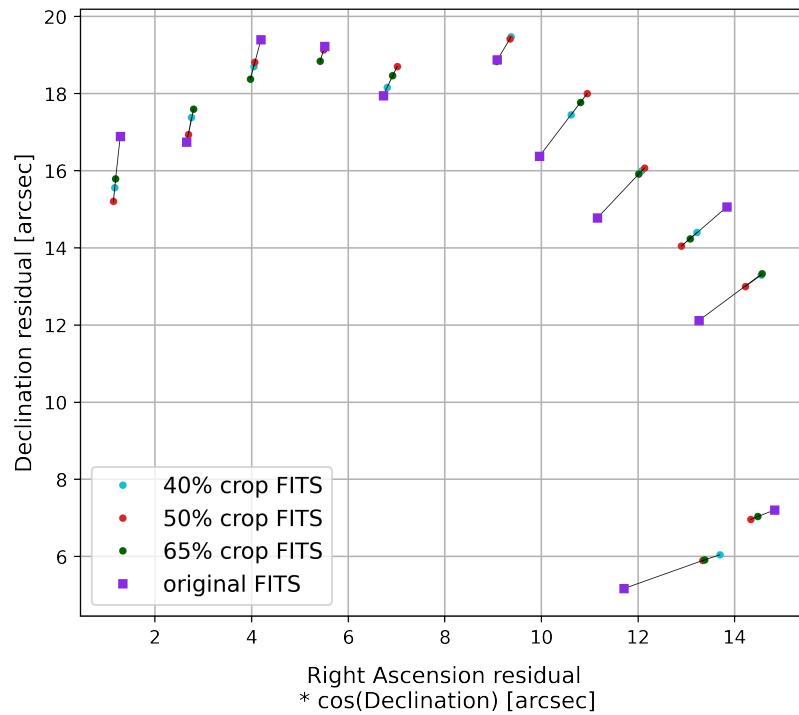


Figure B.25: Centroid RADEC residuals corrected with time bias (campaign 2756)

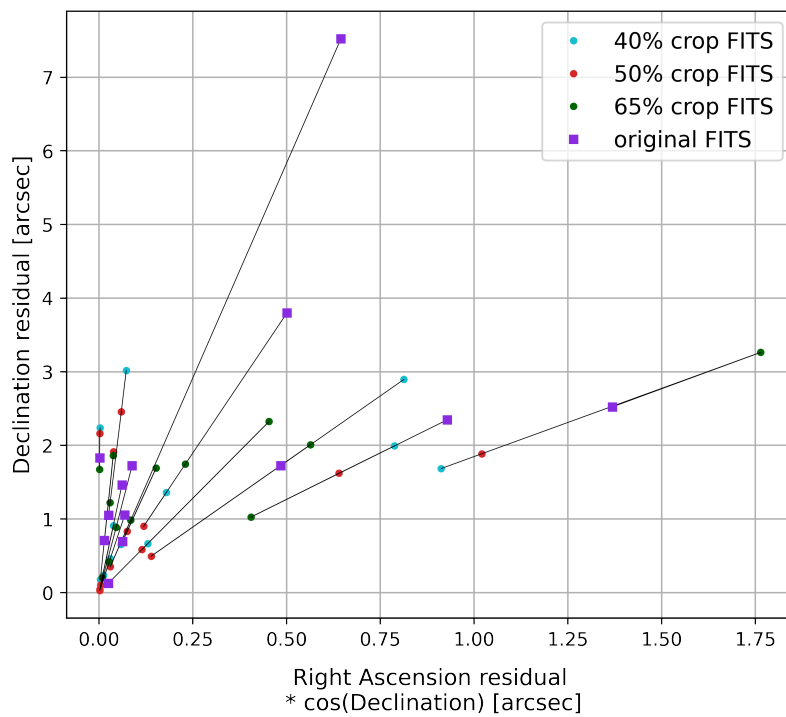


Figure B.26: Centroid RADEC residuals corrected with time bias (campaign 2762)

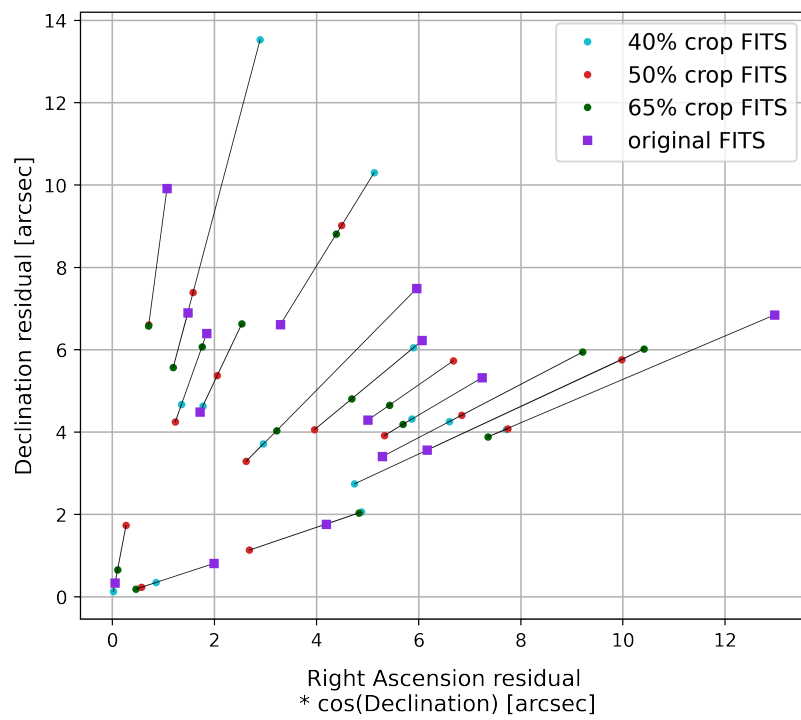


Figure B.27: Centroid RADEC residuals corrected with time bias (campaign 2763)

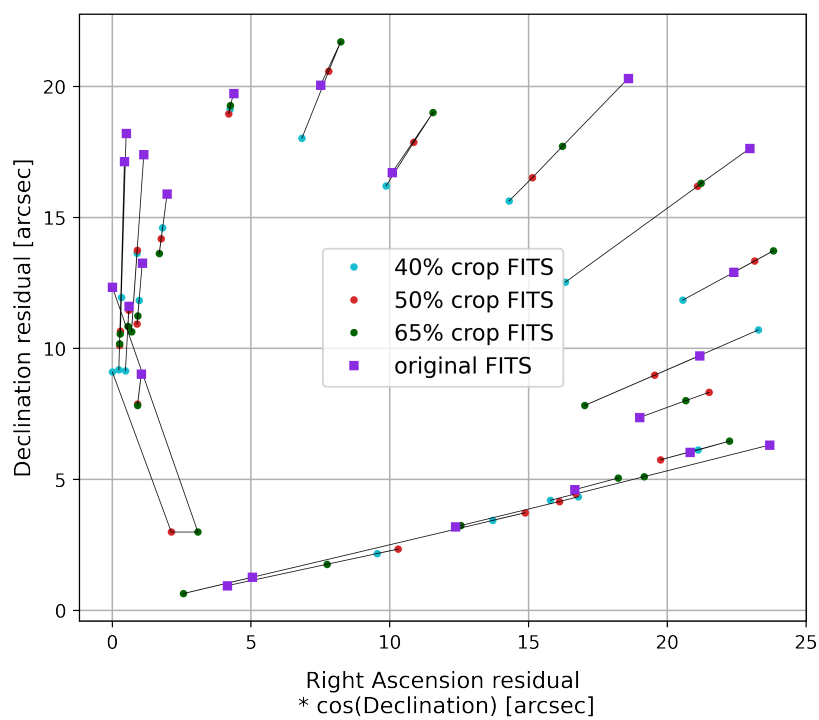


Figure B.28: Centroid RADEC residuals corrected with time bias (campaign 2764)

List of Figures

1.1	Evolution of space objects population	2
1.2	Space Surveillance and Tracking activities	4
1.3	Staring and chasing acquisition modes	5
2.1	Cartesian and spherical coordinates	10
2.2	Earth Centered Inertial frame	11
2.3	Topocentric Equatorial Frame	11
2.4	Topocentric Horizon Frame	11
2.5	Optical sensor scheme	12
2.6	CCD and CMOS architectures	13
2.7	Simplified telescope model	13
2.8	Refractor and reflector telescope	14
2.9	Pratica di Mare Military Telescope for SST	16
2.10	RID functioning scheme	18
2.11	Tracklet detection output	19
2.12	Effect of different image processing	21
2.13	Astrometric calibration example	22
2.14	AstroDART working principle	23
2.15	Tracklet aspect before and after the refinement process	24
3.1	Pipeline integrating RID and AstroDART tools	28
3.2	Analysis of images in PNG format	33
3.3	Analysis of images in FITS format	34
4.1	Effect of size variation of PNG images on pipeline success rate and total computational time (with Scikit-image processing)	39
4.2	Centroid RADEC residuals (Scikit-image processing) - acquisition campaign number 2761	41
4.3	Example of different tracklet identification by ASTRiDE	42
4.4	Pointing error of images in PNG format (Skimage processing)	43
4.5	Different segmentation after the refinement process - FITS vs Skimage PNG	44

4.6	Centroid RADEC residuals corrected with time bias (campaign 2761) - Skimage processing	45
4.7	Effect of size variation of PNG images on pipeline success rate and total computational time (with Matplotlib.pyplot processing)	46
4.8	Centroid RADEC residuals (Matplotlib processing) - acquisition campaign number 2761	48
4.9	Pointing error of images in PNG format (Matplotlib processing)	49
4.10	Different segmentation after the refinement process - FITS vs Matplotlib PNG	50
4.11	Centroid RADEC residuals corrected with time bias (campaign 2761) - Matplotlib processing	51
4.12	Effect of FITS images size variation on pipeline success rate and total computational time	52
4.13	Centroid RADEC residuals (cropped FITS images) - acquisition campaign number 2761	54
4.14	Pointing difference of cropped FITS images	56
4.15	Comparison of pointing and centroid coordinates differences between full-scale and cropped FITS images	57
4.16	Centroid RADEC residuals corrected with time bias (campaign 2761)	58
B.1	Centroid RADEC residuals (Scikit-image processing) - acquisition campaign number 2756	72
B.2	Centroid RADEC residuals (Scikit-image processing) - acquisition campaign number 2762	72
B.3	Centroid RADEC residuals (Scikit-image processing) - acquisition campaign number 2763	73
B.4	Centroid RADEC residuals (Scikit-image processing) - acquisition campaign number 2764	73
B.5	Centroid RADEC residuals corrected with time bias (campaign 2756) - Scikit-image processing	74
B.6	Centroid RADEC residuals corrected with time bias (campaign 2762) - Scikit-image processing	74
B.7	Centroid RADEC residuals corrected with time bias (campaign 2763) - Scikit-image processing	75
B.8	Centroid RADEC residuals corrected with time bias (campaign 2764) - Scikit-image processing	75

B.9 Centroid RADEC residuals (Matplotlib processing) - acquisition campaign number 2756	76
B.10 Centroid RADEC residuals (Matplotlib processing) - acquisition campaign number 2762	76
B.11 Centroid RADEC residuals (Matplotlib processing) - acquisition campaign number 2763	77
B.12 Centroid RADEC residuals (Matplotlib processing) - acquisition campaign number 2764	77
B.13 Centroid RADEC residuals corrected with time bias (campaign 2756) - Matplotlib processing	78
B.14 Centroid RADEC residuals corrected with time bias (campaign 2762) - Matplotlib processing	78
B.15 Centroid RADEC residuals corrected with time bias (campaign 2763) - Matplotlib processing	79
B.16 Centroid RADEC residuals corrected with time bias (campaign 2764) - Matplotlib processing	79
B.17 Centroid RADEC residuals (cropped FITS images) - acquisition campaign number 2756	80
B.18 Centroid RADEC residuals (cropped FITS images) - acquisition campaign number 2762	80
B.19 Centroid RADEC residuals (cropped FITS images) - acquisition campaign number 2763	81
B.20 Centroid RADEC residuals (cropped FITS images) - acquisition campaign number 2764	81
B.21 Comparison of pointing and centroid coordinates differences between full-scale and cropped FITS images (campaign 2756)	82
B.22 Comparison of pointing and centroid coordinates differences between full-scale and cropped FITS images (campaign 2762)	82
B.23 Comparison of pointing and centroid coordinates differences between full-scale and cropped FITS images (campaign 2763)	83
B.24 Comparison of pointing and centroid coordinates differences between full-scale and cropped FITS images (campaign 2764)	83
B.25 Centroid RADEC residuals corrected with time bias (campaign 2756) . . .	84
B.26 Centroid RADEC residuals corrected with time bias (campaign 2762) . . .	84
B.27 Centroid RADEC residuals corrected with time bias (campaign 2763) . . .	85
B.28 Centroid RADEC residuals corrected with time bias (campaign 2764) . . .	85

List of Tables

4.1	Observation campaigns details	38
4.2	<i>deg/pixel</i> ratio variation with resolution	50
4.3	Mean values of RADEC centroid residuals corrected with time bias [arcsec]	59
A.1	Success rate and computational time analyzing images in PNG format with different crop size (Scikit-image processing)	68
A.2	Success rate and computational time analyzing images in PNG format with different crop size (Matplotlib processing)	69
A.3	Success rate and computational time analyzing images in FITS format with different crop size	70

List of Symbols

Variable	Description	SI unit
RA, α	Right Ascension	deg
DEC, δ	Declination	deg
β, θ	Azimuth	deg
ϵ	Elevation	deg
r	Radial distance	m
ϕ	Polar angle	deg
f	Focal length	m
s	Detector size	m
FoV	Field of View	deg
c_x, c_y	Bounding box center coordinates	[-]
w_x, w_y	Bounding box width and height	[-]
O_x, O_y	Image center coordinates	[-]

List of Acronyms and Abbreviations

- ASTRiDE** Automated Streak Detection for Astronomical Images. 6, 22, 23, 27–29, 32–35, 38, 42, 43, 47, 50, 87
- AstroDART** Astronomical Data Analysis and Recovery from Tracklets. 9, 22, 23, 27, 28, 35, 37, 38, 40, 44, 53, 87
- CCD** Charge Coupled Device. 12, 13, 16, 87
- CLAHE** Contrast Limited Adaptive Histogram Equalization. 19
- CMOS** Complementary Metal Oxide Semiconductor. 12, 13, 87
- CNN** Convolutional Neural Network. 17, 18, 29, 33, 40, 45, 53
- CPU** Central Processing Unit. 40
- ECI** Earth Centered Inertial. 10, 11
- ESA** European Space Agency. 1
- FITS** Flexible Image Transport System. 9, 16, 18–23, 27, 29, 32–35, 37, 38, 40, 42–44, 47, 49–59, 70, 80–83, 87–89, 91
- FoV** Field of View. 7, 13, 16, 38, 40, 47, 53
- GEO** Geostationary Earth Orbit. 2
- GPS** Global Positioning System. 16
- GPU** Graphics Processing Unit. 40
- HDU** Header + Data Units. 16
- IAU** International Astronomical Union. 21
- ITAF** Italian Air Force. 15

LEO Low Earth Orbit. 2, 3, 7

ML Machine Learning. 7, 17, 29, 61

NEO Near Earth Object. 6

NWZ North West Zenith. 11

PdM-MiTe Pratica di Mare Military Telescope. 15

PNG Portable Network Graphics. 18, 27, 30, 32–34, 37–40, 42–47, 49–51, 53, 59, 68, 69, 87, 88, 91

PSF Point Spread Function. 5, 6

RADEC Right Ascension and Declination. 22, 25, 29, 32, 34, 35, 37, 40–45, 48–51, 53–55, 58, 59, 72–81, 84, 85, 87–89, 91

RID Real Images Detector. 9, 17, 18, 27–29, 32, 33, 35, 40, 55, 87

SNR Signal to Noise Ratio. 6

SSA Space Situation Awareness. 1, 3

SST Space Surveillance and Tracking. 1, 3, 4, 15, 16, 87

TLE Two-Line Element. 35, 44, 50, 58

UTC Universal Time Coordinated. 17

VM Virtual Machine. 40

WCS World Coordinate System. 21, 22, 25, 32

Acknowledgements

I would like to express my deepest gratitude to all the people that helped me through this path.

First, I would like to thank Professor Pierluigi Di Lizia and Ph.D. Candidates Andrea De Vittori and Riccardo Cipollone, for having inspired this work and guiding me in its development. Thanks for your pieces of advice, your patience and your constant support. Thanks to Luca Facchini for the helpful assistance provided.

Special thanks to Jason, not only for having made available and explained his work, essential for the development of this thesis, but also for having shared with me part of these years of study at PoliMi.

Thanks to all the other fellow students I was lucky enough to share this journey with. In particular, I want to mention Simone, Francesco, Silvia, Marco, Andrea, Lorenzo and Fouad. Each of you helped me along this difficult path and made it more enjoyable to face.

The most important thanks are to my family. Thanks to my uncles, aunts, and cousins who have always supported and encouraged me. Finally, thanks with all my heart to my parents and my brother. Thank you for your sacrifices, and for believing in me. Without your support, this would not have been possible.

