POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Non Linear Multi Target Tracking Near Small Celestial Bodies

Tesi di Laurea Magistrale in
Space Engineering - Ingegneria Spaziale

Author: **Emanuele Marzorati**

Student ID: 225268
Advisor: Prof. Paolo Panicucci
Co-advisors:
Academic Year: 2024-25

Abstract

This thesis investigates the automated tracking of particles ejected from the surface of small celestial bodies, with a particular focus on asteroid Bennu. Inspired by the observations from the OSIRIS-REx mission, where unexpected ejection events were captured by the spacecraft's navigation camera, the study addresses the challenges of particle orbit reconstruction in a constrained observational environment. An extensive review of Bennu's environment, forces acting on particles, and past research was first carried out to support a physically accurate simulation framework. Building on computer vision models and estimation techniques such as the Unscented Kalman Filter, a novel automated tracking algorithm was designed and tested. The algorithm was evaluated using simulations closely resembling real mission conditions, demonstrating strong performance under medium-complexity scenarios but also revealing limitations when tracking multiple targets under cluttered data conditions. This work lays the foundation for future development of tracking algorithms applicable to CubeSats in future exploration missions.

Keywords: Particle Tracking, Small Celestial Bodies, Bennu, Kalman filter, Computer Vision, CubeSats

Abstract in lingua italiana

Questa tesi studia il tracking automatico di particelle espulse dalla superficie di piccoli corpi celesti, con particolare attenzione all'asteroide Bennu. Partendo dalle osservazioni della missione OSIRIS-REx, dove eventi di espulsione sono stati catturati dalla camera di navigazione del veicolo spaziale, lo studio affronta le sfide della ricostruzione dell'orbita delle particelle usando incompleti strumenti di ricerca.
Per creare una simulazione dell'ambiente esterno fisicamente accurata è stata effettuata un'ampia revisione dell'ambiente di Bennu, delle forze che agiscono sulle particelle e delle ricerche precedenti. Basandosi su modelli di Computer Vision e tecniche di stima come l'Unscented Kalman Filter, è stato progettato e testato un nuovo algoritmo di tracking.
L'algoritmo è stato testato utilizzando simulazioni che ricreano le condizioni reali della missione, dimostrando ottime prestazioni in scenari di media complessità, ma rivelando anche i limiti nell'inseguimento di bersagli multipli in condizioni di un gran number di target presenti allo stesso tempo. Questo lavoro pone le basi per lo sviluppo futuro di algoritmi di tracking ai CubeSat in future missioni di esplorazione.

Parole chiave: Particle Tracking, Small Celestial Bodies, Bennu, Kalman filter, Computer Vision, CubeSats

Contents

Introduction

In the last years, the exploration of small celestial bodies like asteroids has become more and more relevant. These objects are considered among the most ancient in the Solar System, and studying them can help us understand how planets formed and how water and organic materials may have arrived on Earth [10, 19, 38]. One of the most interesting discoveries happened during the OSIRIS-REx mission, when scientists noticed that asteroid Bennu was ejecting small particles from its surface [33]. This was unexpected and raised many new scientific questions.

Understanding why and how asteroids eject particles is important for several reasons. From a scientific point of view, these particles can tell us something about the surface and internal structure of the asteroid, its material composition, and how it responds to thermal or mechanical stress [19, 23, 43, 47, 54]. From a mission safety point of view, knowing how particles behave is useful to avoid potential hazards to spacecraft flying nearby [11, 33]. Also, new missions like ESA's Hera will soon approach the Didymos-Dimorphos system, and similar particle ejections are expected there too, especially after the DART impact [28]. This makes Bennu a good test case for preparing algorithms and methods for future scenarios [13, 43].

Another reason why these investigations are interesting is because of the increasing use of CubeSats in planetary missions. These small, cheap satellites often have limited instruments, such as a single monocular camera, and cannot afford advanced sensors like lidars or stereo cameras [48, 65]. This is similar to what happened in OSIRIS-REx: the observations were made with a navigation camera that didn't have depth information [6]. Despite this, the mission was still able to identify and partially study the particle ejections [23, 33]. In future missions, like the one with the Milani CubeSat that will fly with Hera, small platforms will be used again for scientific observations [28, 65]. For these reasons, developing tracking methods that can work under such limited conditions is a relevant goal [16, 36].

Currently, most of the particle tracking done during the OSIRIS-REx mission was based on manual inspection of the images [34]. Automatic methods that exist are usually limited, or they don't scale well when there are many particles or when the measurements are noisy [11, 36, 47]. Also, many tracking algorithms assume that 3D information is available, which is not the case in our scenario [34, 36]. Therefore, the goal of this work is to design and test an automatic particle tracking algorithm that can work even when only monocular images are available, and when the observation data is noisy and sparse [16, 36, 57].

Finally, the current work is built up from a linear approach in tracking detections across image frames, that cannot be considered a feasible alternative in the current environment, where non linearity in the orbits characterize the particle motion around the main gravitational body. The natural engineering choice was to create an algorithm non leveraging stringent linear motion assumptions.

The main research question of this thesis is:

> *How can we design and validate an automated, multi-target particle tracking algorithm that works well with monocular data in low-gravity asteroid environments like Bennu?*

To address this question, the thesis sets the following objectives:

- Review the asteroid environment and the forces acting on ejected particles [43, 53, 54].
- Build a physically consistent simulation environment that includes imaging and motion [36, 43, 47].
- Study and implement estimation techniques such as the Unscented Kalman Filter [16, 24].
- Design and test an automatic algorithm to track multiple particles in image sequences [3, 36, 57].
- Evaluate the performance of the algorithm through simulation and discuss its future applications [16, 36, 57].

This work is structured as follows. Chapter 1 describes the asteroid environment and the ejection phenomena. Chapter 2 presents the necessary tools from computer vision and estimation theory. Chapter 3 describes a basic reconstruction of particle motion. Chapters 4 and 5 focus on the testing and development of the new tracking algorithm. Chapter 6 concludes the work and outlines future developments.

1 | Asteroid Environment

1.1. Introduction

Active asteroids are an uncommon class of celestial objects, grouping small solar system bodies which have an asteroids-like shape and orbit, but display visual evidence of undergoing mass-loss phenomena in the form of dust. If their orbits belong to the area of the solar system between Mars and Jupiter, they are also called main-belt comets. Examples of celestial bodies appertaining to this group, that have been visited by spacecraft missions are the asteroid 101955 Bennu and the dwarf planet Ceres, while the system composed by 65803 Didymos and its moon Dimorphos (formal designation 65803 Didymos I) began showing activity as described before, after the DART collision event. To effectively track the dust particles emitted by the aforementioned celestial bodies, it is essential to first investigate the following foundational aspects:

- The properties of the main body
- The mechanisms of the ejection phenomena
- The characteristics of the emitted particles

To date, the OSIRIS-REx (Origins, Spectral Interpretation, Resource Identification, and Security-Regolith Explorer) mission represents the only comprehensive study of ejection phenomena, having observed multiple ejection events while in orbit around the target asteroid Bennu [1]. This chapter will represent a brief resume of the research already carried out on the environment and characterization of the ejection events, underlying essential concepts that will be used further on in creating a simulation where the ultimate research task will be settled.

[1] Also Ceres is classified as an active asteroid, but the ejecta material in its case is water[27].

1.2. Study of the asteroid Bennu

The composition of the main body is relevant as the ejected particles will likely share common values for properties such as density and albedo. Hereafter, the described properties will be strictly necessary in the measurement process of the particle sizes.
Further on, the internal structure and the mass distribution of the asteroid will dictate the trajectories of the orbits of the revolving particles, making them the base for the gravity field assessment.
The main orbital parameters of the asteroid are listed in table 1.1.

Orbit Element	**Value**
Semimajor Axis $[AU]$	1.1259
Eccentricity $[-]$	0.20372
Inclination $[deg]$	6.034
RAAN $[deg]$	2.017
Argument of periapse $[deg]$	66.304
True anomaly $[deg]$	64.541

Table 1.1: Bennu osculating orbital elements with respect to the ecliptic J2000 frame on March 30, 2019 [9]

1.2.1. Physical properties

The overall shape and gravity field of Bennu were determined, prior to the arrival of the OSIRIS-Rex Mission, through radar and light curve observations, as reported in [45] and [42]. Bennu belongs to the class of rubble pile asteroids, space objects with a diameter of less than 10 km, suspected to be composed by a loose collection of rocks, dust, and smaller fragments held together mainly by gravity and weak cohesive forces.
Asteroids in this category are generally expected to exhibit little to no internal differentiation in mass distribution [55]. However, this was not true the case for Bennu, as investigations into its non-homogeneous mass distribution, conducted through analysis of the gravity field, revealed a highly variable internal density [55], [54]. Consequently the density of Bennu was initially estimated as $2,000 \pm 500$ kgm^{-3} by [12], [19], [38], but in the end the result was refined to $1,190$ kgm^{-3} [10].
The most accurate estimate of the gravitation parameter was produced during the OSIRIS-Rex Mission as 4.8904 ± 0.0009 m^3/s^2, [11]. Because of its its small dimension and mass

the gravity is incredibly weak, generating orbiting velocities in the order of cm/s. The main asteroid physical parameters are listed in table 1.2.

Parameter	**Value**	**1σ uncertainty**
Mass $[kg]$	7.8×10^{10}	0.9×10^{10}
Density $[kg/m^3]$	1190	70
Gravitational parameter μ $[m^3/s^2]$	4.8904 ± 0.0009	0.6
Spin pole RA $J2000$ (α_0) $[deg]$	86.6388	3
Spin pole Dec $J2000$ (β_0) $[deg]$	-65.1086	3
Maximum radius $[m]$	282	5
Mean radius $[m]$	246	10
Polar radius $[m]$	254	26
Rotation period $[hr]$	4.297461	0.002

Table 1.2: Bennu parameters of relevance [22], [45] , [11].

1.3. Main Forces Modeling

Modeling the trajectory of a small particle moving in the Bennu environment requires a detailed model of the forces acting on the particles. In addition to gravity, the effect of radiation pressure on the particles is significant, and cannot be ignored. Even though the current study focus on gravity and solar radiation pressure modeling, in the literature can be found more in depth studies as

- [13], exploiting a three body restricted gravitational representation
- [37], using, on top of the aforementioned gravity model also the Radiation Pressure, Poynting-Robertson Drag, and Solar Wind Drag

The chosen forces reflect the main purpose of these studies. Focusing on the search for relevant figures of merits for the orbits around the asteroid, for example libration points or periodic orbits, the results of the study depend from the simulation of the environment. In contrast, the current research is focused towards the tracking of objects in orbit thanks to computer vision algorithms.
As a reference table 1.3 lists the vast majority of the perturbations and their contribution to the acceleration of bodies orbiting the asteroid.

Source of acceleration	**Acceleration** $[km/s^2]$
Bennu point mass	3×10^{-8}
Bennu gravitational harmonics	6×10^{-10}
Direct solar radiation	4×10^{-10}
Infrared emission from Bennu	5×10^{-11}
Sunlight reflected from Bennu	2×10^{-11}
Unmodeled forces	$< 1 \times 10^{-11}$
Reflected pressure: Direct solar	3×10^{-12}
Reflected pressure: Infrared from Bennu	3×10^{-13}
Reflected pressure: Sunlight from Bennu	3×10^{-13}
Reflected pressure: Sunlight from Bennu	1×10^{-13}
Thermal emission from particle	$< 8 \times 10^{-13}$
Solar tide	5×10^{-14}
Poynting-Robertson Effect	4×10^{-14}

Table 1.3: Forces acting on the particles orbiting Bennu [11]

.

1.3.1. Gravitational Field

Asteroids are small celestial bodies that are usually characterized by irregular shapes, and Bennu is no exception in this regard. Thus, a model based solely on spherical symmetry, not considering its specific shape cannot be considered accurate enough for describing the its gravitational field [49]. Accordingly, the Two-Body Problem [13] and the Homogeneous Rotating Gravitating Triaxial Ellipsoid [7], [29] has to be discarded, while the literature considers as eligible choices the Constant-Density Polyhedron and the Spherical Harmonic Expansion[50]. Following the footsteps of [11], the last mentioned was also the choice of the current study. Subsequently the gravitational potential is modeled as an expansion in spherical harmonics as

$$U = \frac{\mu}{r}\left\{1 + \sum_{n=1}^{\infty}\sum_{m=0}^{n}\left(\frac{R_{\text{ref}}}{r}\right)^{n}\overline{P}_{nm}(\sin\varphi)\left[\overline{C}_{nm}\cos(m\lambda) + \overline{S}_{nm}\sin(m\lambda)\right]\right\} \tag{1.1}$$

Where μ is the product of the universal constant of gravitation G and the mass of the main body M, a_e is the semi-major axis of the main body's reference ellipsoid, and r, φ, λ are the particle's distance, latitude, and longitude, respectively, in a body-fixed coordinate system. C_{nm} and S_{nm} are spherical harmonic coefficients of degree n and order m, and P_{nm} are the associated Legendre functions of degree n and order m, [26]. A gravitational model consists of a set of constants that specify a_e and the C_{nm}, S_{nm} coefficients. In a real world scenario the coefficients are estimated by a series of fly-by around the considered body. The used values are listed in table A.1, these parameters where computed during the OSIRIS-Rex Mission using a number of well-observed particles [11]. In A.1 a complete mathematical derivation of the spherical harmonic expansion gravity can be found. As known, the standard spherical harmonic expansion does not converge globally beneath the main body Brillouin sphere, and in the current scenario this was one of the main topic of concern, as the particles spend a small fraction of their orbit beneath this sphere. Fortunately, it was found that for the particular case of particles orbiting about Bennu, the effect due to divergence was small when compared to truncation error at least up to degree 16, well beyond what can be inferred from the particle detection data [11].

1.3.2. Solar Radiation Pressure

In a micro gravity environment the Solar Radiation Pressure (SRP) is the second most significant force acting on particles orbiting the asteroid, following the asteroid's own gravity. Although weaker than the asteroid's gravitational attraction, SRP is several orders of magnitude stronger than the gravitational influence of the Sun on such particles. SRP acts in the direction opposite to the Sun, influencing the dust grain motion accordingly, and while most models consider the Sun as the sole source of SRP, reflected radiation from nearby celestial surfaces can also contribute.

Moreover, as shown in [23], SRP exhibits seasonal variations, as the observed frequency of particle ejection events tends to decrease from perihelion to aphelion. The cause of this trend has not been clarified already, with the possible decrease in activity with larger distances from the Sun suggest that meteoroid impacts, fracturing of surface boulders due to solar heating, or both may be responsible for ejecting the particles [23].

The extent of SRP's influence depends on the particles' distance from the asteroid, resulting in seasonally stable orbits for some and perturbed orbits for others.

Solar Radiation Pressure (SRP) effects are particularly significant for smaller particles, this is because, for a spherical particle of radius r, the cross-sectional area is $A \ \alpha \ r^2$, while and the mass is $m \ \alpha \ r^3$, thus the ratio $\frac{A}{m} \ \alpha \ \frac{r^2}{r^3} = \frac{1}{r}$.

Numerical simulations presented in [2] indicate that particles in the 0.6–10 cm size range are less influenced by SRP compared to smaller particles, with their dynamics primarily governed by Bennu's gravitational field. However, due to Bennu's weak gravity, SRP cannot be neglected [52].

Furthermore, the limited availability of analytical results [51] in this regime, where both gravity and SRP are simultaneously considered, further complicates the analysis. Under these combined influences, the resulting motion can become chaotic, as discussed in [53].

The equation of motion for a particle subject to SRP is modeled using the cannonball

approximation, assuming a spherical particle shape This is a common simplification, while real particles deviate from sphericity as discussed in [11].
The SRP acceleration is given by the following expression [8, 44]:

$$a_{SRP} = -\frac{Q_{pr} S_0 R_0^2}{c|\mathbf{r}_s - \mathbf{r}|^3} \frac{A}{m} W(\theta, r_\theta)(\mathbf{r}_s - \mathbf{r}) \tag{1.2}$$

Where $\mathbf{r}_s$ is the position vector of the Sun with respect to Bennu, $|\mathbf{r}_s - \mathbf{r}|$ is the distance from the particle's surface element to the Sun, S_0 is the solar constant or radiation flux density at astronomical unit distance R_0, c is the speed of light, and Q_{pr} is the dimensionless efficiency factor for radiation pressure, which depends on the properties (e.g., density, shape, size) of the particle. In the mathematical model, it is assumed to be equal to unity to represent the value of an ideal material.
The values of the SRP parameters used are given in table 1.4.

Parameter	Value	Units	Comments
Q_{pr}	1	–	Efficiency factor
S_0	1.36×10^3	kg/s^3	Solar constant
R_0	1.495978707×10^8	km	Astronomical unit distance
c	$2.99792197442722 \times 10^5$	km/s	Speed of light
ρ	1.190	g/cm^3	Bulk density
r_p	10^{-4}–10	cm	Particles' spherical radius

Table 1.4: SRP Parameters

Shadowing Function

Following [32], the shadowing function $W(\theta, r_\theta)$ is assumed to be cylindrical to simplify the computational time effort of the numerical integration. The cylindrical shadow has a radius value of 290 m, which is close to the polyhedral mesh of Bennu's surface. Its shadowing function angle θ is computed as follows:

$$\zeta = \cos(\theta),$$

where:

$$\zeta = \frac{\mathbf{r} \cdot \mathbf{r}_s}{|\mathbf{r}||\mathbf{r}_s|}$$

and $0 < \theta < \pi$. The shadowing radius r_θ can be computed as:

$$r_\theta = |\mathbf{r}| \sin(\theta)$$

If $\theta > \frac{\pi}{2}$ and $r_\theta > 290\,\mathrm{m}$, then $W(\theta, r_\theta) = 1$. Otherwise, $W(\theta, r_\theta) = 0$.

1.4. Particle Ejection Events

Orbiting around the asteroid Bennu the OSIRIS-Rex mission detected 18 multiparticle ejection events, containing anywhere from a few to 200+ particles. Particles were seen as groups of up to hundreds, radiating from a common source region on the surface of Bennu (ejection events) or as single particles in near-Bennu space. Activity has been observed from a month before perihelion to a month after aphelion, decreasing gently in this span of time lasting 10 months. Despite being investigated in a number of studies, focusing on general cases [4] or specific asteroids, the main cause for the ejection events is still uncertain. The current study does not report this part, as it is not directly related to the main investigated topic.

1.4.1. Observation Bias

The characterization of ejection events is influenced both by the nature of the events themselves and by their observability [23]. The principal sources of bias preventing a deterministic depiction of the events are mostly related to the particle detection capabilities, in particular

- Particles outside the OSIRIS-Rex NavCam 1 FOV and consequently not present in the studies
- Particles too near to Bennu, and so located in the overexposed region of the image surrounding the asteroid, where their spotting becomes more difficult
- Particles in the shadow of Bennu
- Particles behind Bennu
- Particles too faint to be detected because of their location of their size

Moreover, not all the detected particles deliver enough data for their study. In this class of particle is possible to operate a distinction between.

- Particles moving too fast, typically exhibiting hyperbolic orbits, consequently escaping from Bennu's attraction, are not detectable for the minimum number of frames needed to reconstruct a high-fidelity representation of the orbit, mostly because of rate of image capture from the camera [10].

- To the opposite side of the spectrum, particles moving too slow suffer for the same problem of not providing enough data for trajectory estimation [10]. These particles, will likely impact the surface of the asteroid [9].

Lastly, the image cadence, represent a major bottleneck in the orbit characterization, effecting, in the first place, the performances of the employed tracking algorithms, as further investigated in Chapter 4. Afterward, downstream the employment of the tracking algorithm, the orbit determination process requires a sequence with a minimum number of three detections [34]. In the first stages of the mission, when the ejection activity was not known yet or investigated, the image cadence was being set only for navigation purposes, not delivering enough data for scientific purposes. Only in later phases of the mission the image cadence was set to 13 or 19.6 minutes between a couple of images, and even this periods of high activity lasted for a maximum of 16 hours at a time, not representing a consisted portion of the time spent orbiting Bennu [23].

1.4.2. Phase Function

The phase function describes the intensity of sunlight reflected by a particle as a function of the phase angle, defined as the angle subtended by the particle between the Sun and the observer. The determination of the phase function is a prerequisite for estimating particle sizes, as it enables the calculation of the absolute magnitude [2]. The absolute magnitude, H_V, is defined as the apparent V-band magnitude a particle would exhibit if located 1 AU from both the Sun and the observer, observed at a phase angle of 0°. In this studies, photometric data were normalized to particle distances of 1 AU from the Sun and from the observer following the procedure described below[23].

$$V(1,1,\alpha) = m_V - 5 \times \log_{10}(r\Delta) \tag{1.3}$$

Where $V(1,1,\alpha)$ is the normalized magnitude, m_V is the apparent magnitude, r is the particle-Sun distance in AU, and Δ is the particle-observer distance in AU. Phase functions can be modeled in various ways. Due to the limited coverage of phase angles below 70°, the phase function was approximated by a linear fit, yielding the phase coefficient (i.e., the slope of the linear phase function). Phase functions are computed in a number

[2]Although the particle phase function was not directly connected to the present study, it was fundamental to the particle size computation and is therefore reported here.

of studies, table 1.5, resembles the different obtained results through the years.
Remarkably the phase functions of the particles are unlike those seen for asteroid surfaces[23].
Table 1.5 resembles various examples of phase functions computed during the study of Bennu or particles orbiting around it.

Phase Function $[mag/deg]$	**Used Particles**	**Study**	**Study Year**
0.039 ± 0.001	$-$[3]	[21]	2013
0.018 ± 0.005	3	[33]	2019
0.015 ± 0.006	15	[23]	2020

Table 1.5: Particles estimated phase function [23] [4]

Next, to derive H_V values, the phase coefficient 0.015 ± 0.005 mag/deg is extrapolated to $0°$ phase angle.

1.4.3. Particle Size

The photometry derived H_V values are converted into diameters using

$$D_H = \frac{1.329 \times 10^8}{\sqrt{p_V}} 10^{-0.2 H_V} [23] \tag{1.4}$$

where D_H is the particle diameter in centimeters and p_V is the geometric albedo in the V band.
Alternatively, the particle size can also be retrieved from the area to mass ratio η, producing another independent estimate of the particle diameter D_η

$$D_\eta = \frac{150}{\rho\ \eta} \tag{1.5}$$

where ρ is the density of the particle ($kg\ m^{-3}$).
Both methods assume spherical particles, if the particles are spherical and have typical Bennu values for p_V and ρ, then D_H and D_η should agree. Instead, but for a Bennu global mean p_V of 0.044 ± 0.011 and presumed Bennu meteorite density ρ the obtained values for D_H are $\sim$ 1.7× larger than the values of D_η.
Particles on the order of millimeters to centimeters should contain less pore space than

[3] The computation were performed for Bennu itself, not particles.

[4] Where the phase function is defined with the exponent given in the table, plus or minus the associated uncertainty

the bulk asteroid resulting in particle densities greater than the $1,190$ kg m^{-3} bulk density measured for Bennu [23]. The other assumption in the D_H and D_η equations is the sphericity of the particles. Dropping the assumption of sphericity was found that, accounting for a nonspherical shape can lead to consistent size estimates between the D_H and D_η solutions. For an ellipsoid with semiaxes of $a\times$ $a\times$ b, density of $2,000$ kg m^{-3}, and albedo of 0.044, a mean axial ratio $r = \frac{b}{a}$ of 0.27 was found, implying that the particles have a mean albedo and density similar to those measured for Bennu but possess flake-like shapes. The mean axial ratio would be different for other assumed densities and albedos. For example, a less extreme axial ration would support particles with densities less than $2,000$ kg m^{-3} but greater than $1,340$ kg m^{-3} [23].

1.4.4. Particle Production Rate

Through the use of the PSF fitting photometry described above, coupled with the assumption of a geometric albedo of 0.044 and phase coefficient of 0.013 mag/deg, the absolute magnitudes are converted into corresponding diameters expressed in centimeters and masses. It has been estimated that Bennu ejects around 10 kg per orbit or 1.2 years[23]. Lastly, in addition to the previously discussed measurable cases, there are particles simply too small to be detectable from the spacecraft scientific instruments, thus exceeding the lower size detection limit. Assuming an exponential law, these objects represent the vast majority of the population around the asteroid, although accounting only for a small fraction of the detectable mass, [5].

1.4.5. Ejection Event Characterization

As they spanned across multiple month of observation, the different ejection events have been investigated in different papers and by multiple inspection means, also the the different phase angles during observation, together with the before mentioned bias, involve that the events characterization is based on their detectability rather than the properties of the event. The large events on 6 January, 19 January, and 11 February 2019 are discussed in [33], [34] and [47], additional events have been discussed in [23].

[5] Masses are expressed by means of a power law as $y = kx^\alpha$ where k and $alpha$ are constants, and $alpha$ is known as the *power-law exponent*. The notation used in the table distinguish in best estimate, upper and lower bound.

Event Date (2019)	**Number**	**Diameters** $[cm]$	**Velocity** $[m/s]$	**Mass** $[g]$
Jan 06	200+	$0.2 - 5.2$	$0.07 - 3.11$	528^{+3106}_{-353}
Jan 06	200+	$0.2 - 5.0$	$0.07 - 3.26$	459^{+3489}_{-345}
Jan 19	108	$0.5 - 4.3$	$0.06 - 1.29$	231^{+1670}_{-172}
Feb 11	72	$0.3 - 3.8$	$0.07 - 0.21$	223^{+1498}_{-162}
April 19	22	$0.7 - 5.0$	$0.10 - 0.51$	300^{+1451}_{-191}
Sep 13	30	$0.5 - 1.9$	$0.09 - 2.26$	$16^{+124.7}_{-12.3}$

Table 1.6: Greatest Particle Ejection Events Characterization [5]

1.4.6. Orbits

The orbits around Bennu have been described in numerous studies; however, the vast majority are based on simulations [2], [28], [31], with only a few relying on real-world observational data [11], [47]. This paragraph resembles the results of the particle orbit determination process, useful for the environment modeling. The orbit determination methods and pipeline will be described later on in Chapter 2. The particle trajectories can be categorized according to their orbital type in: suborbital, orbital, or escaping. The key parameter that determines into which category a particles falls are the velocity (magnitude and direction) at ejection. Slower velocities will tend toward suborbital and faster velocities will tend toward escape, with the objects experiencing multiple orbital revolutions before impacting falling in between. Inertial or orbital velocities, which are relative to Bennu in a non rotating frame, predominantly range from 10-22 cm/s and show a sharp transition in orbital type around 18-22 cm/s, corresponding to the Bennu escape velocity.

Suborbital Particles

About two thirds of the particles have suborbital trajectories, meaning that they do not survive their first periapsis passage. The apoapsis radius ranges from $\sim$ 275 m (only $\sim$ 40 m above the surface) to $>$ 2.5 km, and the lifetimes range from 0.5 to 26 hours.

Orbital Particles

One fifth of cases are quasi-elliptical orbits with more than one revolution around Bennu. This category can have as few as two revolutions, and in a few cases upward of 10, with the highest number of revolutions being 16.

Escaping Particles

About one seventh of the particle trajectories are hyperbolic with respect to Bennu.

Hyperbolic Flyby Particles

There is the possibility that particles could be ejected on near-parabolic or weakly hyperbolic orbits toward the Sun, in which case SRP could slow and even reverse the particle's escape, sending it back to the near-Bennu environment. In this case, particles are subject to an expulsion, and after a time, falling back towards the asteroid performing a flyby. Despite predicted in simulations [43], trajectories of this type has not been documented yet.

2 | Fundamentals of Computer Vision and Estimation

2.1. Introduction

This chapter represents an introduction to models and methods that will be used during the course of this work.
For clarity and completeness, all the mathematical models, related to computer vision and to estimation theory, that will be used to perform operations further on, will be already presented here. The contextualization of these algorithm, how they will be ordered to compose the complete measurement pipeline implemented, will be the subject of future dissertations.
Firstly, a description on how real objects creates an image on a digital sensor will take place. Following will be discussed how to retrieve an estimate of the position of an object in real world coordinate system using pixel coordinates only.
Lastly, fundamental concepts of the estimation theory such as the Kalman Filter, with some of its derivations, will be outlined.

2.2. The Pinhole Camera Model

Let's consider the design of a basic camera system capable of capturing an image of a three-dimensional (3D) scene or object. One of the most straightforward designs involves placing a barrier with a tiny opening, called an aperture, between the 3D scene and a light-sensitive surface such as photographic film or a digital sensor. Each point on the 3D object emits many rays of light in all directions. Without the barrier, light from various points in the scene would reach every location on the film, resulting in a blurred image.
However, with the barrier in place, only a single light ray (or a narrow bundle of rays) from each point on the 3D object can pass through the small aperture and reach the film. This setup establishes a unique correspondence between each point in the 3D scene and a point on the film. The outcome is the formation of an image on the film that faithfully

represents the 3D scene. This concept forms the basis of what is known as the pinhole camera model.

2.2.1. Formalizing the Pinhole Camera Model

A more structured depiction of the pinhole camera model is shown in Figure 2.1. In this setup, the light-sensitive surface is typically called the image plane or retinal plane. The small opening in the barrier is referred to as the pinhole (denoted by O), and serves as the optical center of the camera. The perpendicular distance from the image plane to the pinhole is known as the focal length, denoted by f.
To understand how a point in space is projected, let's denote a 3D point as $P = [x, y, z]$. Through the pinhole, this point is projected onto the image plane at $P = [x, y]^T$. The pinhole itself can also be mapped onto the image plane as a reference point C.
A local coordinate system (i, j, k) can be defined as centered at the pinhole O. In this camera coordinate system, the axis k is oriented perpendicular to the image plane and points toward it. The line connecting the pinhole O and its image projection C is known as the optical axis of the camera system.

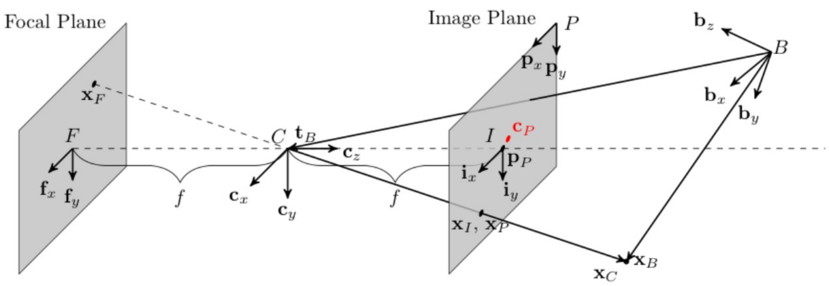


Figure 2.1: Pinhole Camera Model Reference Frames [35]

Recalling that point P' is derived from the projection of 3D point P on the Π image plane. Therefore, can be derived a relationship between 3D point P and image plane point P'. Notice that triangle $P^{'}C^{'}O$ is similar to the triangle formed by P, O and $(0, 0, z)$. Therefore, using the law of similar triangles it is found that

$$P^{'} = \begin{bmatrix} x' & y' \end{bmatrix} = \begin{bmatrix} f\frac{x}{z} & f\frac{y}{z} \end{bmatrix}^T \tag{2.1}$$

2.2.2. Introduction to the Camera Matrix Model

Introducing a set of new parameters it is possible to express the the camera model in matrix form. The first parameters, c_x and c_y, describe how image plane and digital image coordinates can differ by a translation. Image plane coordinates have their origin C at the image center where the k axis intersects the image plane. On the other hand, digital image coordinates typically have their origin at the lower-left corner of the image. Thus, the 2D points in the image plane are defined by a translation vector $\begin{bmatrix} c_x & cy \end{bmatrix}^T$. To accommodate this change of coordinate systems, the mapping now becomes:

$$P' = \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} f\frac{x}{z} + p_x \\ f\frac{y}{z} + p_y \end{bmatrix} \tag{2.2}$$

The second effect it has to be account for is that the points in digital images are expressed in pixels, while points in image plane are represented in physical measurements (e.g. centimeters). In order to accommodate this change of units, we must introduce two new parameters k and l. These parameters, whose unit of measure is $pixels \times cm$, correspond to the change of units in the two axes of the image plane. Note that k and l may be different because the aspect ratio of a pixel is not guaranteed to be one. If k = l, it is said that the camera has square pixels. Adjusting the previous mapping accordingly

$$P' = \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} fk\frac{x}{z} + p_x \\ fl\frac{y}{z} + p_y \end{bmatrix} = \begin{bmatrix} k_x\frac{x}{z} + p_x \\ k_y\frac{y}{z} + p_y \end{bmatrix} \tag{2.3}$$

2.2.3. Homogeneous Coordinates

The projection from the last equation is not linear, this problem is solvable changing coordinates system. A new coordinate is introduces such as any point $P' = (x', y')$ becomes $P' = (x', y', 1)$. Similarly, any point $P = (x, y, z)$ becomes $(x, y, z, 1)$. This augmented space is referred to as the homogeneous coordinate system. Adjusting the formulation using homogeneous coordinates it is possible to formulate [20]

$$P'_h = \begin{bmatrix} k_x\frac{x}{z} + p_x \\ k_y\frac{y}{z} + p_y \\ z \end{bmatrix} = \begin{bmatrix} k_x & 0 & p_x & 0 \\ 0 & k_y & p_y & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{2.4}$$

From the previous equation the intrisic camera matrix can be extrapolated as

$$\mathbf{K} = \begin{bmatrix} k_x & 0 & p_x \\ 0 & k_y & p_y \end{bmatrix} \tag{2.5}$$

This represent a linear map from real world coordinates $(x, y, z) \in R^3$ to pixel coordinates $(x'_{pix}, y'_{pix}) \in R^2$

2.3. Inverse Projection of the Pinhole Camera Model

Finding a way to reverse the pinhole camera model, solving the problem of 3D objects placement for monocular vision, has produced for a lot of years a certain number of academic studies. This was caused by the the system itself being under determined, admitting infinitely many solutions in R^3, having to solve a system characterized by three unknowns but only two equations. In order to supersede the complete lost of the variable representing the depth in the field of view a lot of ways have been prescribed.

2.3.1. Line Of Sight Vector

One way of solving this problem leverages the line of sight vector, defined as

[Line of Sight Vector] Let $\mathbf{p} = [u, v]^T$ be a pixel coordinate in the image plane, and let $K \in \mathbb{R}^{3\times3}$ be the intrinsic camera matrix. The *line of sight (LOS) vector* $\mathbf{l} \in \mathbb{R}^3$ corresponding to $\mathbf{p}$ is the direction in 3D space from the camera center through the pixel $\mathbf{p}$, and is given by:

$$\mathbf{l} = K^{-1}\tilde{\mathbf{p}},$$

where $\tilde{\mathbf{p}} = [u, v, 1]^T$ is the homogeneous representation of the pixel coordinate. The LOS vector defines a ray in 3D space and provides the direction of the point corresponding to $\mathbf{p}$, with unknown depth.

Because the final result is lacking the range information, in contrast with the complete determination of the position in a 3D space, the system of equations computing the line of sight vector admits only a determinate solution. In order to mediate this problem, multiple line of sight vectors are used to determine the position of the target. Because in space the tracked object usually has a relative velocity with respect to the observer it is simply unfeasible to work with multiple measurements taken with the same camera at the same object, each separated by a delta time, while the observer is moving. Two found solutions are the usage of

- Sterovision-based estimation, where two cameras are mounted on the same observer satellite tracking another object [1], [18]
- Vision-based relative navigation, a technique developed for satellites constellations in which an high number of line of sight measurements, from one satellite to the others are used to create a determinate system whose solution are the ranges between the probes [56]

Both this techniques are not applicable in the studied case, where the measurements are coming from a single camera. The reasons for finding a solution also for this type of configurations are dictated by the increasing popularity of CubeSats also for scientific purposes in interplanetary missions.

2.3.2. Neural Networks

Lastly, many studies present a depth estimator that leverages neural networks. In contrast with the previously described one, these algorithms work on a monocular image but unfortunately, they are all using as determining assumptions on how the colors, shadows or other visual attributes are varying in the image according to its change in depth [64], [46]. Making them ineligible choices in an environment where these features cannot be known a priori.

2.3.3. GIANT

GIANT Goddard Image Analysis and Navigation Tool) is an API developed by NASA (National Aeronautics and Space Administration), and in particular by A. J. Liounis (who also kindly provided the references for the download [1]) to perform most Optical Navigation tasks including geometric camera calibration, attitude estimation, unresolved body center finding, resolved body center finding, limb based navigation, surface feature navigation and image planning. Its employment in the study of the ejecta phenomena will be described further on.
Here it is described how the algorithm provides an estimate of the target inertial position given its position on the camera sensor, thus pixel coordinates, along with the orientation and position of the observer camera.
A mathematical derivation of the line of sight vector follows.
The intrinsic matrix inverse is computed analytically as

[1] https://software.nasa.gov/software/GSC-18758-1

$$d = k_x k_y - k_{xy} k_{yx} \tag{2.6}$$

$$\mathbf{K}^{-1} = \begin{bmatrix} \frac{k_y}{d} & -\frac{k_{xy}}{d} & \frac{k_{xy}p_y - k_y p_x}{d} \\ -\frac{k_{yx}}{d} & \frac{k_x}{d} & \frac{k_{yx}p_x - k_x p_y}{d} \end{bmatrix} \tag{2.7}$$

Line of Sight Vector

Firstly, the pixel locations x_P are converted units of distance and re-centered at the principal point by multiplying by the inverse of intrinsic matrix.

$$\mathbf{x}'_I = \mathbf{K}^{-1} \begin{bmatrix} \mathbf{x}_P \\ 1 \end{bmatrix} \tag{2.8}$$

$$\mathbf{K}^{-1} = \begin{bmatrix} \frac{1}{k_x} & 0 & \frac{-p_x}{k_x} \\ 0 & \frac{1}{k_y} & \frac{-p_y}{k_y} \end{bmatrix} \tag{2.9}$$

This is the step by step computation

$$\mathbf{x}'_I = \left(\mathbf{K}^{-1}_{[:,:2]} \cdot \mathbf{x}_P\right)^T + (\mathbf{K}^{-1}_{[:,2]})^T \tag{2.10}$$

Where:

$$\mathbf{K}^{-1} = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \end{bmatrix} \tag{2.11}$$

The first two columns:

$$\mathbf{K}^{-1}_{[:,:2]} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \tag{2.12}$$

The third column:

$$\mathbf{K}^{-1}_{[:,2]} = \begin{bmatrix} k_{13} \\ k_{23} \end{bmatrix} \tag{2.13}$$

The pixel vector:

$$\mathbf{x}_P = \begin{bmatrix} p_x \\ p_y \end{bmatrix} \tag{2.14}$$

Multiplication result:

$$\mathbf{K}^{-1}_{[:,:2]} \cdot \mathbf{x}_P = \begin{bmatrix} k_{11}p_x + k_{12}p_y \\ k_{21}p_x + k_{22}p_y \end{bmatrix} \tag{2.15}$$

Adding the third column:

$$\mathbf{x}'_I = \begin{bmatrix} k_{11}p_x + k_{12}p_y + k_{13} \\ k_{21}p_x + k_{22}p_y + k_{23} \end{bmatrix} \tag{2.16}$$

This result being called gnomic locations. Once the gnomic locations are retrieved, the units vectors are formed according to

$$\mathbf{x}_C = \begin{bmatrix} \mathbf{x}_I \\ f \end{bmatrix} \tag{2.17}$$

Finally the line of sight is computed by normalizing the vector

$$\hat{\mathbf{x}}_C = \frac{\mathbf{x}_C}{\|\mathbf{x}_C\|} \tag{2.18}$$

2.4. Kalman Filters

Filtering is an operation that involves extraction of information about a quantity of interest x_k at (discrete) time k by using data measured up to, and including time k. Therefore, the objective of filtering is to recursively estimate x_k (target state) from the measurements z_k.

The Kalman Filter [25] represents a solution of the classical filtering and prediction problem, providing estimates of the parameters of interest from a series of noised measurements, that are modeled as random processes.

In contrast to batch filters, such as the least squares method and all its derivations, the Kalman Filter can be summarized as as a sequential iterative prediction-correction process, making it suitable in the processing of real time data [40].

In contrast, batch filters are not employable in the current work for the need of a sequential estimation approach to be applied in an actual space mission on board a satellite. However, the standard Kalman filter, being essentially a linear estimator, is not suitable to be applied for non-linear systems, or when the nonlinearities become too important to be neglected. In order to solve this problem, non-linear replacements, such as, the Extended Kalman Filter (EKF), and the Unscented Kalman Filter (UKF) have been derived, and will be here briefly presented.

2.4.1. Kalman Filter State and Measurements

The target state vector is $\underline{x}_k \in R^n$, where k is the time index. The target space evolves according to the discrete time stochastic model

$$\underline{x}_k = \phi_{k-1}(\underline{x}_{k-1}, \underline{u}_{k-1}) \tag{2.19}$$

Where ϕ_{k-1} is a known, linear function of state $\underline{x}_{k-1}$, while $\underline{u}_{k-1}$ is the noise, accounting for mis-modelling or disturbances in the target motion.
The measurements of the process are

$$\underline{y}_k = h_k(\underline{x}_k, \underline{\epsilon}_k) \tag{2.20}$$

So when at time t_k a new measurement is acquired

$$\delta\underline{y}_k = \tilde{H}_k \delta\underline{x}_k + \delta\underline{\epsilon}_k \tag{2.21}$$

Where $\tilde{H}_k$ is a known, possibly non linear, function and $\underline{\epsilon}_k$ is the measurement noise, assumed to be a gaussian random vector with zero mean and assigned covariance.

$$E[\underline{\epsilon}_k] = \begin{bmatrix} E[\underline{\epsilon}_{k1}] \\ \vdots \\ E[\underline{\epsilon}_{kl}] \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \tag{2.22}$$

$$E[\underline{\epsilon}_k \underline{\epsilon}^T_k] = \begin{bmatrix} R_{11} & R_{12} & \dots & R_{1l} \\ R_{12}^T & R_{22} & & \\ \vdots & & \ddots & \\ R_{1l}^T & & & R_{ll} \end{bmatrix} = \mathbf{R} \tag{2.23}$$

Where $\mathbf{R}$ is the noise matrix.
The matrix $\mathbf{R}$ is usually diagonal, with $\mathbf{R}_{ij} = 0$ if $i \neq j$. In contrast, when the measurements are correlated, this is no longer the case, and correlations can arise between terms, for example, between observed angles and ranges. Additionally, the noise matrix can account for time correlations between measurements. In real-world scenarios, the matrix $\mathbf{R}$ may also vary over time, for instance, due to changes in the size of the measured object within the field of view. Nevertheless, in this case, it is assumed that the matrix $\mathbf{R}$, and

thus the measurement error, remain constant.

2.4.2. Extended Kalman Filter

The principal drawback of the Kalman Filter is its reliance on the linearization of the reference trajectory, restricting its application for linear scenarios only.
In contrast, the EKF mitigates the loss of accuracy changing the reference trajectory at each observation. With the intrinsic edge of having a linearization closer and closer to the true trajectory as the filter ingests observations.

Filtering Algorithm

In order to propagate the filter for the generic step from time $t_{k-1} \rightarrow t_k$, the used quantities are

- The a-posteriori information of the previous step, mean state $\hat{x}^+_{k-1}$ and covariance P^+_{k-1} at time t_{k-1}
- The measurements at time t_k: $\underline{y}_k$ with noise matrix $\mathbf{R}_k$

The necessary computations

- Integrate from t_{k-1} to t_k

$$\begin{cases} \dot{\underline{x}}^* = \underline{f}(\underline{x}^*, t) \\ \dot{\Phi} = A(t)\Phi \end{cases} \tag{2.24}$$

  Using as initial conditions

$$\begin{cases} \underline{x}^*(t_{k-1}) = \hat{\underline{x}}^+_{k-1} \\ \Phi(t_{k-1}, t_{k-1}) = I \end{cases} \tag{2.25}$$

  Where $\dot{\underline{x}}^*$ is the reference trajectory and Φ is the state transition matrix, defined as the jacobian of the flow

$$\frac{\partial \varphi}{\partial \underline{x}_0} = \begin{bmatrix} \frac{\partial \varphi_x}{\partial x_0} & \frac{\partial \varphi_x}{\partial y_0} & \cdots & \frac{\partial \varphi_x}{\partial v_{z0}} \\ \vdots & & & \\ \frac{\partial \varphi_{vz}}{\partial x_0} & \frac{\partial \varphi_{vz}}{\partial y_0} & \cdots & \frac{\partial \varphi_{vz}}{\partial v_{z0}} \end{bmatrix} \tag{2.26}$$

  And the notation used is

$$\frac{\partial \varphi}{\partial \underline{x}_0}(\underline{x}_0, t_0, t) = \Phi(\underline{x}_0, t_0, t) = \Phi(t_0, t) \tag{2.27}$$

- Perform the prediction step

$$\delta \underline{\hat{x}}_k^- = \Phi(t_{k-1}, t_k)\delta \underline{\hat{x}}_{k-1}^+ \tag{2.28}$$

$$P_k^- = \Phi(t_{k-1}, t_k)P_{k-1}^+\Phi^T(t_{k-1}, t_k) \tag{2.29}$$

- Compute

$$\delta \underline{y}_k = \underline{y}_k - \underline{h}(\underline{x}_k^*, t_k) \tag{2.30}$$

$$\tilde{H}_k = \frac{\partial \underline{h}}{\partial \underline{x}}(\partial \underline{x}_k^*, t_k) \tag{2.31}$$

- Perform the update step
  - Compute the Kalman gain $K_k = P_k^- \tilde{H}_k^T \left[\tilde{H}_k P_k^- \tilde{H}_k^T + R_k\right]^{-1}$
  - Compute the a-posteriori mean and covariance

$$\delta \underline{\hat{x}}_k^+ = \delta \underline{\hat{x}}_k^- + K_k \left[\delta \underline{y}_k - \tilde{H}_k \underline{\hat{x}}_k^-\right] \tag{2.32}$$

$$P_k^+ = \left[I - K_k \tilde{H}_k\right] P_k^- \tag{2.33}$$

The first expression becoming

$$\delta \underline{\hat{x}}_k^+ = K_k \left[\delta \underline{y}_k\right] \tag{2.34}$$

To enforce symmetry P_k^+ is computed as $P_k^+ = \frac{P_k^+ + (P_k^+)^T}{2}$

- Update the reference trajectory

$$\underline{\hat{x}}_k^+ = \underline{x}_k^+ + \delta \underline{\hat{x}}_k^+ \tag{2.35}$$

Replace K with $k+1$ and repeat from the first step until all measurements have been ingested

2.4.3. Unscented Kalman Filter

The idea of the UKF is to use the Unscented Transform (UT) to predict mean and covariance of the measured state. Along with mean and covariance of the predicted measurements, these quantities will be used to compute the residuals of the measurements at time t_k.

Unscented Transform

The Unscented Transform [24], [60] is a way to perform the propagation in time of a probability density function, in the process known as uncertainty propagation. It is usually employed in case of non linear dynamics. The UT does not represent correctly the entire final probability density function, focusing instead on computing correctly only its mean and covariance. This is achieved by categorizing the initial probability density function using a small number of samples, whose main task is to preserve the initial distribution, these will be propagated using a nonlinear map. Finally the final mean and covariance are estimated from the propagated sigma points.
Given the non linear transformation $\underline{y} = \underline{g}(\underline{x})$, given $\hat{\underline{x}}$ and $\mathbf{P}_x$, the most general way to evaluate the sigma points comprehends

- Compute $\sqrt{(n+\lambda)\mathbf{P}_x}$ using for example the Cholesky decomposition
- Compose the sigma points as a column vector of

$$\begin{aligned}
\underline{\chi}_0 &= \hat{\underline{x}} \\
\underline{\chi}_i &= \hat{\underline{x}} + \left(\sqrt{(n+\lambda)\mathbf{P}_x}\right)_i \quad \text{for } i = 1, \ldots, n \\
\underline{\chi}_i &= \hat{\underline{x}} - \left(\sqrt{(n+\lambda)\mathbf{P}_x}\right)_{i-n} \quad \text{for } i = n+1, \ldots, 2n
\end{aligned} \tag{2.36}$$

- Where
  - The scaling parameter is computed as $\lambda = \alpha^2(n+k) - n$
  - The spread of the sigma points is determined by α
  - And $k = 0$
- Compute the weights as

$$\begin{aligned}
W_0^{(m)} &= \frac{\lambda}{(n+\lambda)} \\
W_0^{(0)} &= \frac{\lambda}{(n+\lambda)} + (1 - \alpha^2 + \beta) \\
W_i^{(m)} &= W_i^{(0)} = \frac{1}{[2(n+\lambda)]} \quad \text{for } i = 1, \ldots, 2n
\end{aligned} \tag{2.37}$$

With $\beta = 2$

Filtering Algorithm

In order to propagate the filter for the generic step from time $t_{k-1} \rightarrow t_k$, the used quantities are

- The mean state $\hat{x}^+_{k-1}$ and the covariance P^+_{k-1} at time t_{k-1}
- Using the measurements at time t_k: $\underline{y}_k$ with noise matrix $\mathbf{R}_k$

The necessary computations

1. Compute the sigma points $\underline{\chi}_{i,k-1}$ and weights $W_i^{(m)}, W_i^{(0)}$
2. Perform the prediction step
   - Propagate the sigma points from t_{k-1} to t_k, obtaining $\underline{\chi}_{i,k}$
   - Compute the a priori mean state as

$$\underline{\hat{x}}_k^- = \sum_{i=0}^{2l} W_i^{(m)} \underline{\chi}_{i,k} \tag{2.38}$$

   - Compute the a priori covariance as

$$P_k^- = \sum_{i=0}^{2l} W_i^{(0)} \left[\underline{\chi}_{i,k} - \underline{\hat{\chi}}_{i,k}^-\right] \left[\underline{\chi}_{i,k} - \underline{\hat{\chi}}_{i,k}^-\right]^T \tag{2.39}$$

   - Compute the sigma points in measurement space

$$\underline{Y}_{i,k} = \underline{h}(\underline{Y}_{i,k}) \tag{2.40}$$

   - Compute the mean of the measurements

$$\underline{\hat{y}}_k = \sum_{i=0}^{2l} W_i^{(m)} \underline{Y}_{i,k} \tag{2.41}$$

3. Perform the update step

- Compute the measurements covariance and cross correlation with the state

$$
\begin{aligned}
P_{YY,K} &= \sum_{i=0}^{2L} W_i^0 \left[\underline{Y}_{i,k} - \underline{\hat{y}}_k\right] \left[\underline{Y}_{i,k} - \underline{\hat{y}}_k\right]^T + \mathbf{R}_k \\
P_{XY,K} &= \sum_{i=0}^{2L} W_i^0 \left[\underline{\chi}_{i,k} - \underline{\hat{x}}_k^-\right] \left[\underline{Y}_{i,k} - \underline{\hat{y}}_k\right]^T
\end{aligned}
\tag{2.42}
$$

- Compute the Kalman Gain $K_k = P_{xy,k} P_{yy,k}^{-1}$
- Compute the a posteriori mean and covariance

$$
\begin{aligned}
\underline{\hat{x}}_k^+ &= \underline{\hat{x}}_k^- + \mathbf{K}_k(\underline{y}_k - \underline{\hat{y}}_k) \\
P_k^+ &= P_k^- - \mathbf{K}_k P_{yy,k} \mathbf{K}_k^T
\end{aligned}
\tag{2.43}
$$

2.4.4. Multi-Hypothesis Tracking Algorithm Overview

Developed to track multiple targets in a cluttered environment, and renown for its performances in space applications [3], the Multiple Hypothesis Tracking (MHT) algorithm combines multiple Kalman Filters (KF) in a single framework. It builds a tree of potential track hypotheses for each candidate target, thereby providing a systematic solution to the data association problem. In this approach, each time a new measurement is received, the algorithm evaluates the probability that the measurement came from a previously known target in a database, from a new target, or that it is a false detection. These probabilities are calculated and updated recursively as more measurements are received over time.

Each hypothesis corresponds to a specific data-association scenario, and for every hypothesis, the target states are estimated using a Kalman filter. This enables the algorithm to refine the estimates based on the likelihood of each association. As new data comes in, the set of possible hypotheses grows, and the algorithm builds a branching structure to manage all possible measurement-to-track associations.

The main challenge in applying MHT is correctly associating measurements with their corresponding targets, especially in environments with high clutter or noise. To manage this complexity, the framework often relies on clustering techniques. Clustering is the process of dividing the full set of targets and measurements into smaller, independent groups (or clusters), which helps reduce the computational burden and allows for more efficient hypothesis evaluation.

The hypothesis generation method used in this implementation is measurement-oriented. This means that, for each incoming measurement, a set of hypotheses is generated to

cover all possible sources, rather than predicting measurements from known targets (as in target-oriented approaches). While this allows for a thorough treatment of all possibilities, it also leads to a significant increase in the number of hypotheses to evaluate.
This growth in hypothesis count is the main computational drawback of MHT. As the number of potential associations increases, so does the computational load required to process them. As stated in [21], the success of MHT strongly depends on its ability to manage and maintain a reduced, meaningful list of hypotheses through techniques like pruning and merging. Despite this challenge, MHT has shown high computational efficiency in practice [57], making it a valuable method for multi-target tracking in complex environments.

3 | Particle Ejection Events Reconstruction

3.1. Introduction

In order to reconstruct the orbits of the ejected particles, the first fundamental step is to link, from one image to the next ones, the detections belonging to the same particle. This initial pairing of measurements is called tracking.
It is worth to be highlighted that manual inspection [36], i.e. the recognition of tracks by a human operator, visualizing the photos taken by the OSIRIS-Rex NavCam aided only by an ad hoc post processing of the images, played a major role in the study of the ejection events, and it is still considered the ground truth as of today.
Nevertheless, the literature has produced different methods to perform this task in an autonomous way, mainly because of the high volume of photos produced in later stages of the mission when the study of the ejection events became of main relevance. For the case study of Bennu two algorithms were produced. The two algorithms are fundamentally opposed in terms of their initial assumptions and, consequently, in the categories of particles with which they perform optimally. As a result, they are well-suited to be applied to the same set of images in a complementary manner. In particular, the first described algorithm is designed to be applied on the data not recognized by the second algorithm. The second depicted algorithm is not leveraging linear motion approximations, resembling much more the final work outlined in this study, nonetheless future improvements could be achieved by integrating components of the first algorithm into the second.

3.2. Linear Approximation of the Particle Motion

This algorithm is applied to the particles in the images, whose detections appear as streaks of pixels because of their high velocity. From one side, a streak of pixels represents multiple couples of x and y pixels measurements, consecutively producing more information than multiple, not already connected, sparse, point like detections. On the other hand, unfortunately, particles traveling this fast stay in the camera field of view (FOV) only for a brief amount of time, consequently producing less information that can be used for the orbit determination process, and ultimately making it more challenging. During the mission, the image acquisition pipeline, performed by the OSIRIS-Rex NavCam, prescribes two photos, separated only by a brief amount of time. The acquirement of always a couple of photos holds true for each possible image cadence, i.e. the time interval between two couples of photos.

3.2.1. Detections Identification

Subsequently, all the photos, containing detections as described before, are corrected for distortion and registered on the center of Bennu.
The complete post processing pipeline of the photos taken by the OSIRIS-Rex NavCam can be found in [36]. This algorithm, despite its paramount importance for the study of the particles around Bennu, is very different from the pipeline implemented in the current research, mainly because of the assumption of the particles traveling with linear motion assumption in the camera FOV. Nevertheless, here are being reported the aspects useful to the current investigation.
If a streak is present in both the photos of a couple, it can be considered as a four-epoch track, one position at the beginning and one at the end of each image's exposure.

3.2.2. Radiant Point Estimation

The pipeline leverages the assumption that all the particles are ejected from the same point on the surface of the asteroid, the radiant point. A complete mathematical derivation can be found in B. Based on the two observations available for each particle, the radiant point shared by all particles in the image can be estimated by assuming that they were all ejected from a single origin and that their velocities remained constant in both magnitude and direction after ejection, resulting in straight-line motion across the image.
In this approach, all images were registered with respect to the center of Bennu, hereafter the radiant point was estimated by applying a least squares method to determine the intersection of the lines defined by each pair of particle observations. The 1σ uncertainty was computed as the root-mean-square distance between the estimated radiant point and the nearest approach of each line. However, the assumption of straight-line particle motion introduced a bias in the estimated radiant point, influenced by the particle velocities and the elapsed time between ejection and the first observation.

3.2.3. Ejection Epoch Estimation

When only two observations (one image) are available the ejection epoch and the orientation of the ejecta cone produce an indeterminate system. Thus, the 3-D states to be related to the particles are undetermined. In order to solve this problem it was considered that the trajectories were initially constrained to the image plane, and therefore, the observation angle rate was constrained to be constant. This causes particles traveling toward or away from the spacecraft to predict later or earlier event times, respectively. Consequently, the time of the event was taken as the median of the times estimated using each individual particle, and the standard deviation of the times was used for the 1σ uncertainty. Doing so it is assumed that the median particle is traveling within the image plane, producing the ejecta cone centering around the image plane. Given the lack of any connection between the image plane and the ejection characteristic the made assumption is not necessarily true, but it was considered as a reasonable approximation.
In case the same object was captured in each one of the couple image, it becomes possible, using three observations to find a deterministic solution of the ejecta cone orientation. The method compares the lengths of the two successive streaks. If the streak is longer in the second image, it means that the particle is traveling toward the camera at constant velocity, and if the streak is shorter in the second image, this is interpreted as the particle traveling away from the camera at constant velocity.

3.2.4. Ejection Location Estimation and Uncertainty

The radiant point found following the prescribed manner corresponds to two different points on the surface of Bennu, the first being on the side of Bennu near the camera, while the second one being of the opposite side of Bennu, out of the camera FOV. To find the solution corresponding to the nearest ejecta point, rays originating from the spacecraft's position at the acquisition time of the first image were projected toward the estimated radiant point, and their intersection with the Bennu shape model was computed using the asteroid's rotational state corresponding to the inferred event time. This was performed using the tools of the GIANT toolkit.

3.2.5. Three-Dimensional Object States

The three-dimensional motion of particles was inferred by contrasting the measured rate of change in observation angle with the theoretical rate expected for a particle located at the same angular distance from the radiant, assuming it was ejected at the estimated event time and moved strictly within the image plane (normal to the camera's boresight).

Particles exhibiting a faster rate than predicted were considered to be approaching the observer, while those with a slower rate were interpreted as moving away. This approach continued to assume straight-line trajectories without acceleration and a common ejection time for all particles.

3.2.6. Results

The algorithm is applied to the fastest particles emitted from Bennu, consequently the results are an estimate of the fastest speeds the ejected particles can achieve. This value is of main relevance as the precision achieved by the second algorithm, even tough not designed to be applied to the fastest particles, depends mostly on the speed shown by the particles and as it will be shown in 4 is not particularly robust for enhance values of particle velocities. The measured ejection velocities are grouped in the table below.

Event Date (2019)	**Number**	**Min Vel.** $[m/s]$	**Mean Vel.** $[m/s]$	**Max Vel.** $[m/s]$
Jan 06	117	0.069	0.464	3.111
Jan 19	13	0.482	0.866	1.221
Feb 11	30	0.086	0.151	0.511

Table 3.1: Particle Ejection Events Characterization, High Speed Particles [47]

3.3. Automated Tracking of Particles Across Image Frames

This algorithms represents the nearest source to the current work, after being described, and it results shown, it will undergo an accurate performance analysis in Chapter 4 were the validity of the employed assumptions will be tested.
The pseudo-code for this algorithm is presented in 3.1.

3.3.1. Automated Tracking Algorithm Overview

To perform automated particle tracking, a multiple object tracking strategy is used to simultaneously fit and link particle trajectories across consecutive image frames. The process iterates over each image in the sequence: at each step, for a given frame k, the algorithm seeks to establish new trajectories that originate in that image, ignoring those already initiated in earlier frames. This is done using an Extended Kalman Filter (EKF), which predicts future particle positions based on the sequence of observations from image

k up to image $k+l$.

Each possible forward trajectory gives rise to a separate EKF instance. For instance, if a particle is detected in Image 1 and has two potential matches in Image 2, two separate EKFs are initialized, one for each match, and propagated forward to Image 3. At this point, a data association step identifies additional candidates. If the first EKF finds three possible matches and the second finds two, each filter is duplicated to follow all valid options, resulting in five EKFs carried forward to Image 4. This propagation continues either until the final frame is reached or no further viable associations can be made under the pairing algorithm's rules.

Once each EKF reaches its terminal point, backward smoothing is applied to determine whether the trajectory is physical consistent, i.e. it is a feasible orbit. This is assessed by evaluating the post-fit standard deviation of the residuals. Only those filters meeting the consistency threshold are accepted, and the corresponding observations are removed from the candidate pool. The same process is then repeated, this time starting from Image 2.

3.3.2. Initial Pairing

At the beginning of processing each new image, the correct associations between detections in the current frame and those in future frames are unknown. Additionally, the motion of the particles that produced the current detections is not yet determined. As a result, a broad set of candidate pairings is initially permitted to account for this uncertainty. For this preliminary step, it is assumed that each particle responsible for a detection lies along its line-of-sight vector, positioned at the same distance from the camera as the surface of Bennu. Furthermore, it is presumed that the particle remains stationary, i.e. their velocity is zero in the inertial frame centered and fixed to the asteroid, allowing the use of known camera poses, positions, and intrinsic parameters to project the particle's inertial position into future image frames.

In each subsequent image, the Euclidean distance is calculated between the projected location of the particle from the initial image and the detected particles in that later image. In the described literature approach candidate pairings are then accepted if this distance falls below a threshold: 400 pixels for sparse fields and 200 pixels for dense fields [36]. The large association threshold was chosen because the authors knew that the particles were not stationary [36]. This initial association process is conducted for all images captured within one hour after the reference image. At this stage, any detections already linked to trajectories originating in earlier frames are excluded from consideration.

3.3.3. EKF Design and Tuning

Once the initial pairing, between detection in the first considered image and subsequent one, have been performed, an EKF is used to follow these potential tracks and determine whether more observations are linkable to them or not.

EKF Dynamics

Tracking objects that are orbiting a small body, with a small gravity field, it was chosen to model the gravity field of the central body and the solar radiation pressure. The gravity is modeled by treating Bennu as a point mass, while a cannonball model is employed for the solar radiation pressure. The used values come from [54].

EKF State Vector

The state vector contains includes the inertial position and velocity of the particle with respect to Bennu, as well as a solar radiation term to estimate the effects of solar radiation on the particles, leading to

$$\mathbf{x} = [p_x \, p_y \, p_z \, v_x \, v_y \, v_z \, s]$$

The solar radiation term represents the combination of the surface area of the object, the coefficient of reflection, and the mass of the object

$$s = \frac{c_r A}{m}$$

where c_r is the coefficient of reflection, A is the surface area of the particle, and m is the mass of the particle. To initialize the state vector, we assume that the object is located along the line-of-sight vector through the observation at the same distance as Bennu is from the camera.

EKF Measurements

The EKF updates are performed using particle observations in each image, represented in pixel coordinates. The transformation of the particle state into predicted observations at a given time is carried out using the known camera attitude and its geometric model.

Subsequent Detection Pairings

Once the initial detection pairs are established, an EKF is initiated for each pair to forecast the particle's location in subsequent images. These EKF predictions are then compared to the actual detections in the next frame using both Euclidean and Mahalanobis distance metrics [39]. A detection is considered a valid match if it falls within the defined thresholds for both distance measures. For every such match, a new EKF is created by cloning the originating EKF—duplicating its current state, covariance, and all previously assimilated measurements. The matched detection is then incorporated as the next measurement in the clone, and the tracking process continues iteratively. Notably, the Euclidean distance threshold used to determine valid pairings is adaptive: as the EKF assimilates more observations and improves its prediction accuracy, the threshold is progressively reduced.

$$d_{thresh} = 50(8 - n) + 10[36] \tag{3.1}$$

where d_{thresh} is the distance threshold in pixels, n is the number of measurements that have been processed by the EKF, and d_{thresh} has a minimum value of 10 pixels [36]. The Mahalanobis threshold instead is defined as

$$m^2 = (\mathbf{y} - \hat{\mathbf{y}})^T(\mathbf{P}_x^I + \mathbf{R})^{-1}(\mathbf{y} - \hat{\mathbf{y}}) \tag{3.2}$$

with m^2 being the squared Mahalanobis distance, $\mathbf{y}$ being the pixel location of the current observation under consideration, $\hat{\mathbf{y}}$ being the predicted location of the observation in the current image, $\mathbf{P}_x^I$ being the covariance of the state vector mapped into the measurement space, and $\mathbf{R}$ being the covariance matrix of the observation itself in the measurement space. The Mahalanbis threshold was set to 25. This process is repeated until every EKF started from the first image has either run out of future images to consider or has reached a point where no more potential pairs meet the distance thresholds. Figure 3.1 shows a graphical representation of the algorithm.

Backward Smoothing, Identification of Valid Tracks, and Next Images

After the pairing phase concludes, either because each EKF initialized from the first image has reached the final frame or no future detections satisfy the distance thresholds, each EKF undergoes a backward smoothing procedure to determine the optimal trajectory across all associated measurements. The resulting residuals from this smoothing step are evaluated to identify which EKFs successfully traced valid particle paths. Tracks exhibiting smoothed residuals with a standard deviation below 25 pixels (1σ) are classified as valid tracks [36]. The detections linked to these validated tracks are then excluded from the pool of candidate detections, and the entire process is repeated starting from the next image, continuing until all images have been processed.

Algorithm 3.1 Automated Tracking Algorithm

```
for each image index k = 1 to N do
  Extract unassigned observations O_k from image I_k
  for each observation o_k in O_k do
    for each subsequent image index t = k + 1 to N do
      Extract unassigned observations O_t in I_t
      for each observation o_t in O_t do
        Initial Particle Association
        if o_t linkable to o_k then
          Check number of already paired detections
          if First found pair then
            Initialize EKF
          else
            Propagate EKF
            if Multiple Connection Possible then
              Clone EKF
            end if
          end if
        end if
      end for
      Backward Smoothing of Tracks
    end for
  end for
```

3.4. Summary

In this chapter are being summed up the techniques already employed in determining the orbits of the particle orbiting around a small body. The two techniques are presented as part to the same pipeline, working in a complementary manner on different parts of the same set of data. The results obtained were subsequently shown. The importance of the second algorithm, not leveraging a linear motion assumption, and chosen for this reason, as a base for further developments has been discussed.

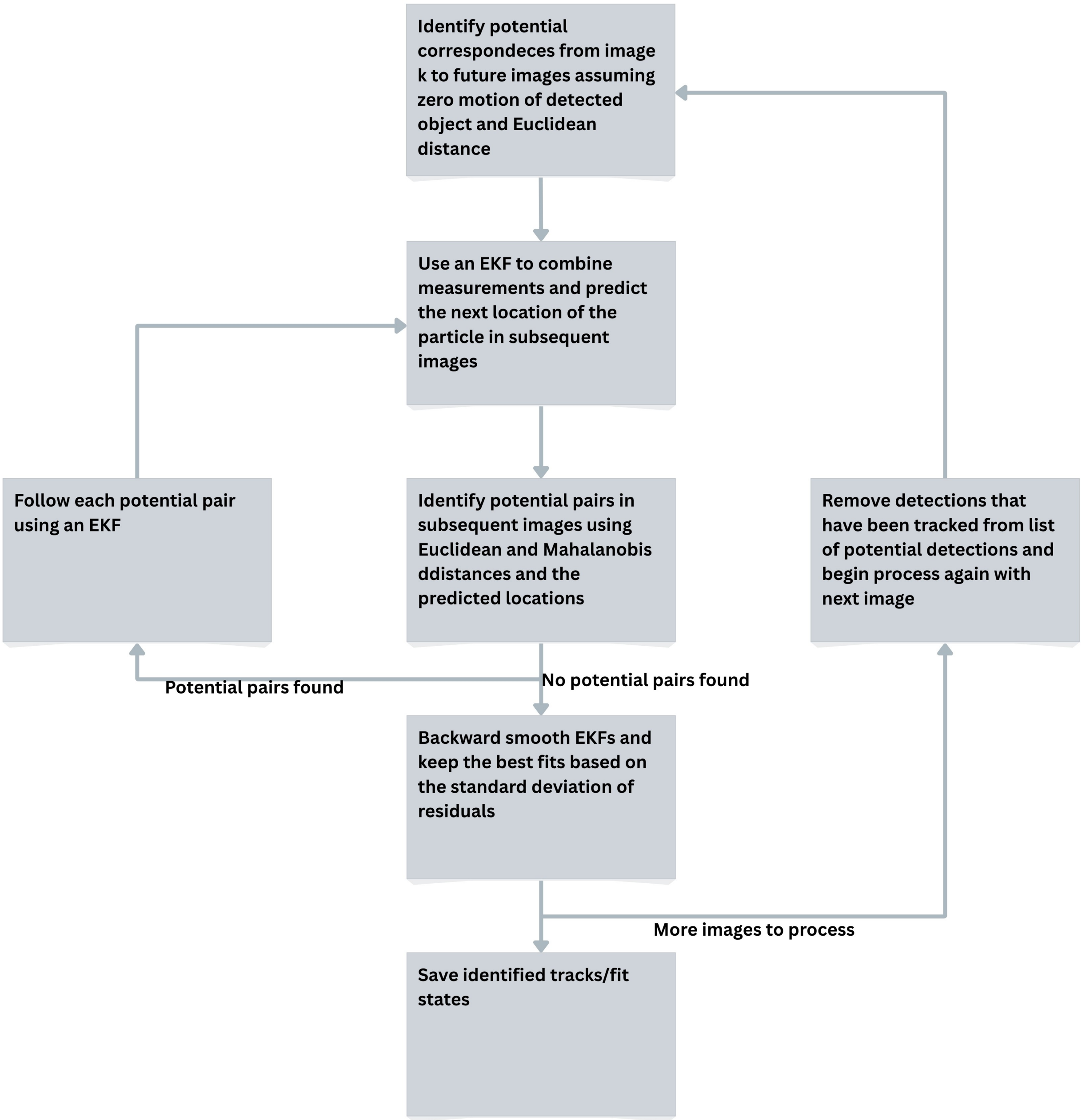


Figure 3.1: The process of the automated tracking algorithm., Canva Generated

4 | Automated Tracking Algorithm Testing

4.1. Introduction

Since the previously described Automated Tracking Algorithm represents the most accurate example of a tracking algorithm around particle near the asteroid Bennu the literature has produced, an in depth review and testing of its implemented features, fundamental for the particle tracking, was carried out. The objectives of this analysis were:

- Understand how particular tasks, that do not have as of today a definitive solution in the literature were carried out
- Test the algorithm robustness to variations of the external conditions, understanding the limits of applicability of its pipeline to different environments and ejecta events, with the ultimate scope of evaluating its robustness, along with the capability for being adopted to other systems

4.2. Testing Platform Implementation

In order to evaluate the capabilities of the tracking algorithm, an accurate modeling of the environment where the implemented pipeline will be employed, is mandatory. Consequently, the orbiting environment near the asteroid Bennu as well as the ejection events themselves must be correctly recreated. Moreover, in regards to the implementation of the simulated measurement acquisition pipeline, the described modeling principles hold true.

4.2.1. Environment Simulation

The asteroid's orbital environment is simulated in MATLAB, the values and coefficients used can be found in Chapter 1, more specifically the tables: 1.1, 1.2, 1.3, 1.4.

Orbit of the asteroid Bennu

In the various simulations that have been carried out, the considered orbit of the asteroid around the Sun has been retrieved using the data contained in the planetary ephemeris kernel of PSpice. At the very start, the vast majority of them were settled in the time span between January and February 2019, to resemble as close as possible the framework of the most documented particle ejection events of January 06, January 19 and February 11. Subsequently, after the initial testing of the simulating platform, the time span was enlarged covering as many orbit positions as possible, to inquire the influence of the distance of the asteroid from the Sun. The result of this first inspection showed that from the evolution of the orbits it was not possible to understand any visible variation. Nevertheless the module of the acceleration given by the Solar Radiation Pressure was computed along the orbit as showed in Figure 4.1.

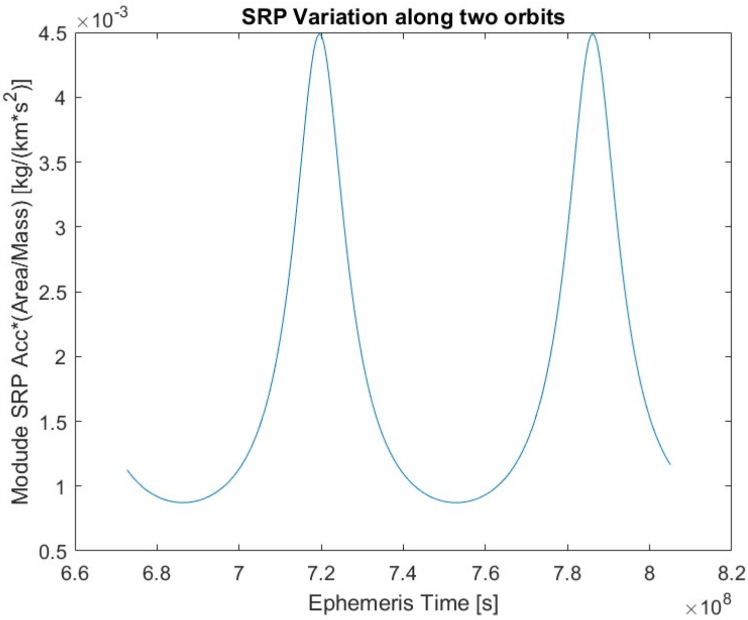


Figure 4.1: SRP Variation Across Two Orbits, Matlab Simulation

In Figure 4.2 the orbits of Bennu and other relevant celestial bodies are shown.

Orbits around the asteroid Bennu

In order to propagate the orbits of the ejected particles around Bennu a spherical harmonics gravity model and an SRP cannon-ball model were implemented, the equations of motion were composed as described in A.1. In order to test the validity of such model, the produced orbits were then tested comparing the recreated orbits with the ones derived from the ephemeris of the OSIRIS-Rex probe revolving around the asteroid, derived

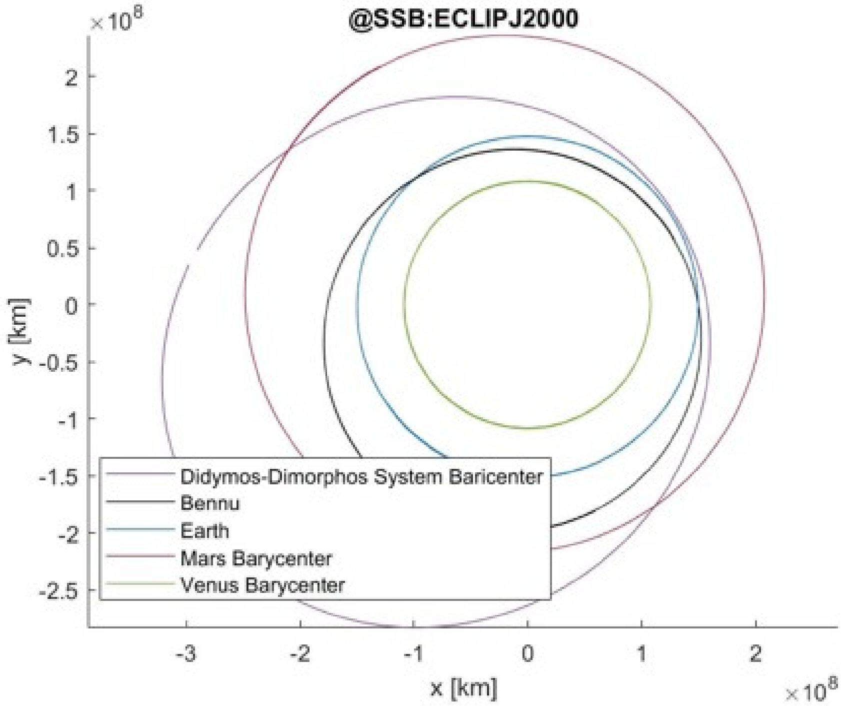
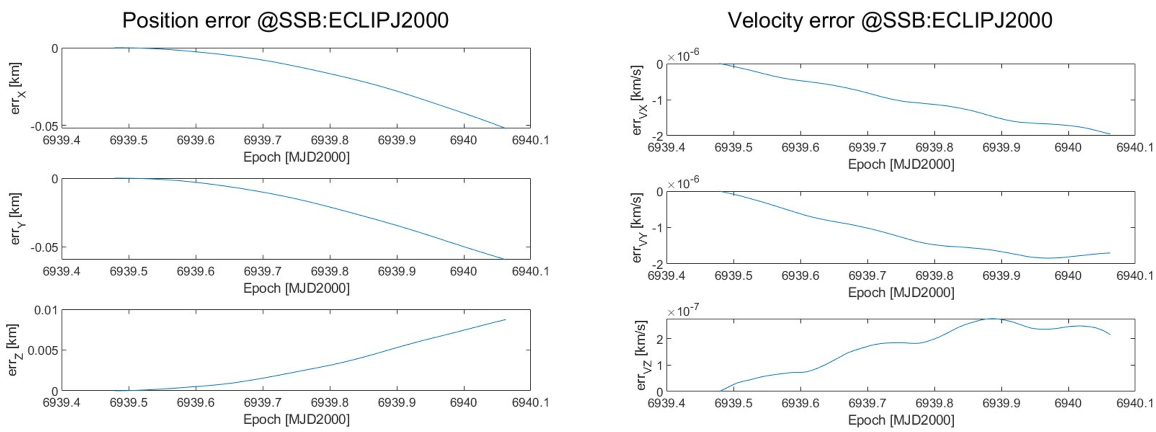


Figure 4.2: Bennu orbit in the solar system, MATLAB simulation

from the corresponding PSpice kernel, the result in terms of position and velocity error is shown in Figure 4.3.



(a) Position Error

(b) Velocity Error

Figure 4.3: Examples of simulated particle trajectories

The considered time span was varied as mentioned above. Subsequently, the particles propagated trajectories were retained, while the propagated orbits of the probe were discarded, employing the ephemeris trajectory for the spacecraft position.

Properties of the Ejected Particles

The traits of the ejected particle resembles closely Table 1.6, the diameters in particular are generated from a Gaussian distribution, defined as:

$$f(x \mid \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{4.1}$$

where $f(x \mid \mu, \sigma)$ is the probability density function, μ is the mean and σ is the standard deviation.
To create the particles from the Gaussian distribution, a mean and a standard deviation has to be specified. To choose these values the minimum and maximum value for the diameters were set, according to Table 1.6, to 0.08 cm and 7 cm respectively. From these values the mean radius was computed as $r_{mean} = \frac{r_{min}+r_{max}}{2}$ and the standard deviation $\sigma = \frac{r_{max}-r_{min}}{6}$. Subsequently, the volumes of the generated particles are computed.

Ejection Events

The particle ejection events reconstruction starts with generating a feasible initial state for the particles to then be propagated. The used initial states can be divided in two categories.

- Particles that are kicked in orbit from the asteroid's surface, thus, having initial position on the ellipsoid having as axes the radii of the asteroids, the shape model not being employed because of hardware limitations. Their initial velocity, magnitude and directions, decides if they perform an hyperbolic escape or fall back towards the asteroid, see Subfigure 4.4a.
- Particles whose initial state is in the space nearby the asteroid, characterized by a certain orbital radius and velocity components, see Subfigure 4.4b.

In the first case to perform the attribution of a state given a particle the subsequent pipeline is followed.
The particle position is generated considering spherical coordinates adapted to an ellipsoid as following

$$x = a \sin\theta \cos\phi$$
$$y = b \sin\theta \sin\phi$$
$$z = c \cos\theta$$

Where $a,\ b,\ c$ are the ellipsoid semi-major axes, θ is the polar angle, the angle from the

positive x-axis, $0 \leq \theta \leq \pi$, ϕ is the azimuthal angle, the angle from the positive x-axis in the xy-plane, $0 \leq \phi < 2\pi$. And the angles are generated from a normal distribution.
Given the volume of a particle, its mass is computed considering as density the estimated particle density. Subsequently, knowing the average kinetic energy for each ejection, the velocity modulus is computed. The resulting velocities were investigated, resulting similar in range and magnitude, when compared to the true ones. In order to establish the components of the velocity the previously spherical ejection angles, θ and ϕ are considered. The normal distribution used to extract the velocity angles has the same center, i.e. the mean of the angles θ and ϕ, while its variance is increased. The particle velocity takes the form of

$$\begin{aligned} v_x &= |v| \sin\theta \cos\phi \\ v_y &= |v| \sin\theta \sin\phi \\ v_z &= |v| \cos\theta \end{aligned}$$

For each trajectory of the orbiting particles, the orbital elements are sampled from continuous uniform distributions, with one distribution corresponding to each orbital element. In this case, the probability density function is

$$f(x) = \begin{cases} \frac{1}{b-a} & for\ a \leq x \leq b \\ 0 & for\ x \leq a\ or\ x \geq b \end{cases} \tag{4.2}$$

The choice of two different probability density functions reflects the intention to model two distinct physical scenarios. In the first case, the particles originate from a single ejection event, resulting in a concentrated distribution around a central region, with density gradually decreasing towards the edges. In the second case, the particles are the outcome of entirely uncorrelated events, leading to a sparse and scattered orbital distribution, with no apparent relationship among their trajectories. The ranges used for the osculating orbital elements are listed in Table 4.1, while the results of various simulations can be found in Figure 4.4.

4.2.2. Retrieving the Measurements

The process of simulating the retrievement of the measurements taken by the probe, while orbiting the asteroid, tries to mimic as much as possible what occurs in the real world.

[1] Where r_{Bennu} is the average of the three asteroid radii given by CSpice

Orbit Element	Value
Semi-Major Axis $[r_{Bennu}]^1$	$2-8$
Eccentricity $[-]$	$0.5-0.97$
Inclination $[rad]$	$0-2\pi$
RAAN $[rad]$	$0-2\pi$
Argument of periapse $[rad]$	$0-2\pi$
Initial True anomaly $[rad]$	$0-2\pi$

Table 4.1: Osculating orbital elements, with respect to the ecliptic J2000 frame, of the particle orbiting Bennu

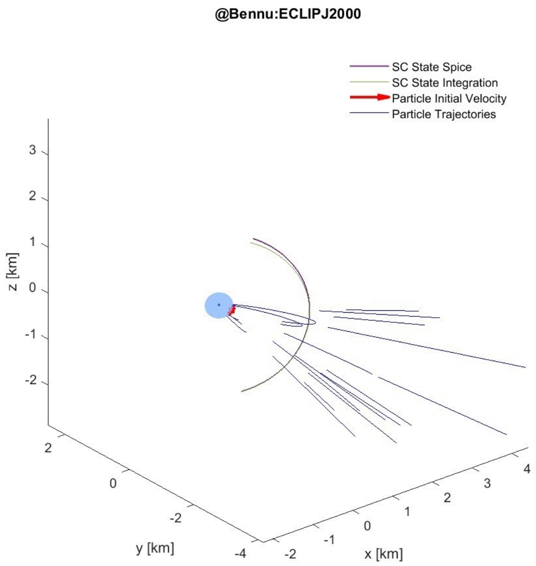
Firstly, considering the image cadence of the OSIRIS-Rex NavCam during the mission, the shooting time instants are computed in the simulation time span. Then, once both the particles trajectories and the spacecraft's are computed, for each shooting time, the particles states, are converted from inertial to the local vertical, local horizontal frame (LVLH).

Transformation to Local Vertical Local Horizontal Frame

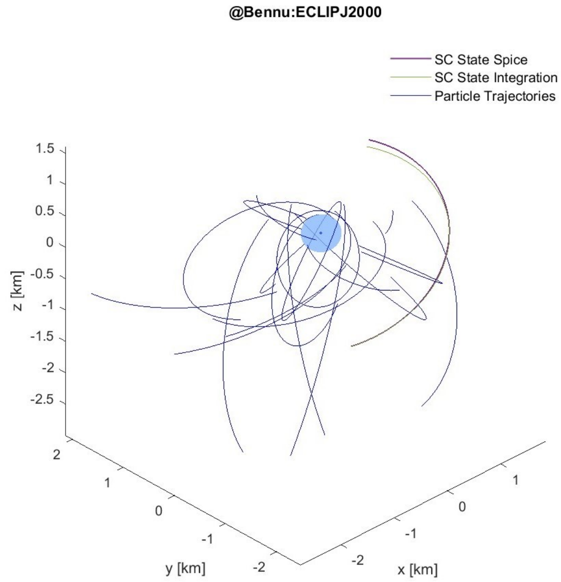
This frame is centered on the spacecraft, has the radial, x-axis, pointing towards the center of the central body, the z-axis perpendicular to the orbital plane, following the angular momentum vector direction and the y-axis completing the right-handed system, pointing in the direction of motion. The coordinates transformation is performed both for position and velocity as

$$\vec{x}_{particle\ LVLH} = RotMat_{LVLH}\ (\vec{x}_{particle\ IN}\ - \vec{Offset}) \tag{4.3}$$

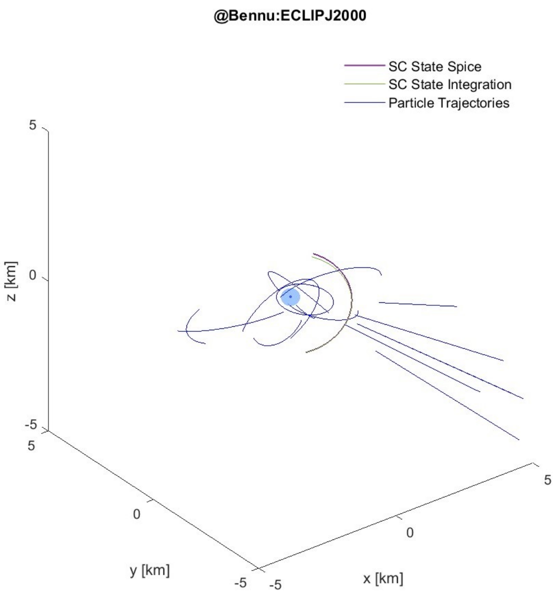
Where $\vec{x}_{particle\ LVLH}$ is the state of the particle in Local Vertical, Local Horizontal Reference Frame, $\vec{x}_{particle\ IN}$ is the state of the particle in inertial frame. The offset, the translation between the origins of two coordinate frames, is defined, in this case, as the state vector of the spacecraft.



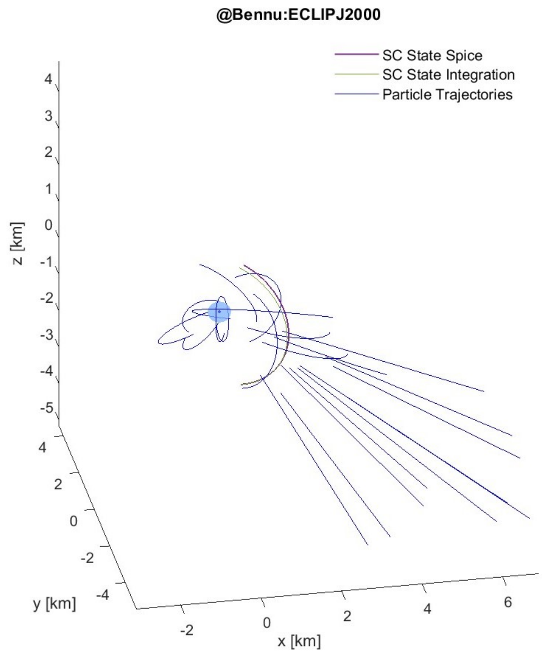
(a) Particles Ejected from the surface



(b) Particles orbiting the asteroid



(c) Particles Orbiting + Ejected Total Number 10



(d) Particles Orbiting + Ejected Total Number 20

Figure 4.4: Examples of simulated particle trajectories

Computation of Rotation Matrix from Inertial to Local Vertical Local Horizontal reference frame

Given, the position vector of the satellite, $\vec{r}$, the velocity vector of the satellite, $\vec{v}$, time, t and the gravitational parameter $\mu = GM$

- Compute Norms and Angular Momentum:

$$r = ||\vec{r}||$$
$$\vec{h} = \vec{r} \times \vec{v}$$
$$h = ||\vec{h}||$$
$$\vec{a} = \vec{v}t = \texttt{RHS}(t, [\vec{r}; \vec{v}], GM)_{(4:6)}{}^2$$

- Define the LVLH Frame, the rotation matrix R from the inertial frame to the local orbital frame is constructed using the orthonormal basis vectors:

$$\hat{i} = \frac{\vec{r}}{\vec{r}}, \quad \hat{k} = \frac{\vec{h}}{\vec{h}}, \quad \hat{j} = \hat{k} \times \hat{i}$$

So the rotation matrix is:

$$R = \begin{bmatrix} \hat{i}^T \\ \hat{j}^T \\ \hat{k}^T \end{bmatrix}$$

- Compute Time Derivatives of the Basis Vectors

$$\hat{i}_t = \frac{1}{r}\left(\vec{v} - \left(\hat{i} \cdot \vec{v}\right)\hat{i}\right)$$
$$\hat{k}_t = \frac{1}{h}\left(\vec{r} \times \vec{a} - \left(\hat{k} \cdot (\vec{r} \times \vec{a})\right)\hat{k}\right)$$
$$\hat{j}_t = \hat{k}t \times \hat{i} + \hat{k} \times \hat{i}t$$

The time derivative of the rotation matrix is then:

$$\dot{R} = \begin{bmatrix} \hat{i}_t^T \\ \hat{j}_t^T \\ \hat{k}_t^T \end{bmatrix}$$

- Assemble the Full Rotation Matrix as a 6×6 rotation matrix used in state-space transformations (e.g., linearized dynamics)

$$\text{Rot\_Matrix} = \begin{bmatrix} R & 0_{3\times3} \\ \dot{R} & R \end{bmatrix}$$

This matrix transforms position and velocity vectors from the inertial frame to the rotating LVLH frame, including the effect of the rotating basis on velocity.

Other Rotations

From the LVLH reference frame the particles positions are rotated in order to comply with the used reference frame of the pinhole camera model, shown in 4.5.

$$r_{PinHole} = R_{\text{Pin}} R_{\text{Cam}} r_{LVLH} \tag{4.4}$$

Where

- The first rotation produces a frame, called Camera Reference Frame, characterized by the x-axis looking in the spacecraft's forward direction and the y-axis facing the radial direction. The used matrix

$$R_{\text{Cam}} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- The second rotation aligns the axis with the reference frame adopted by the implemented Pinhole Camera Model

$$R_{\text{Pin}} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix}$$

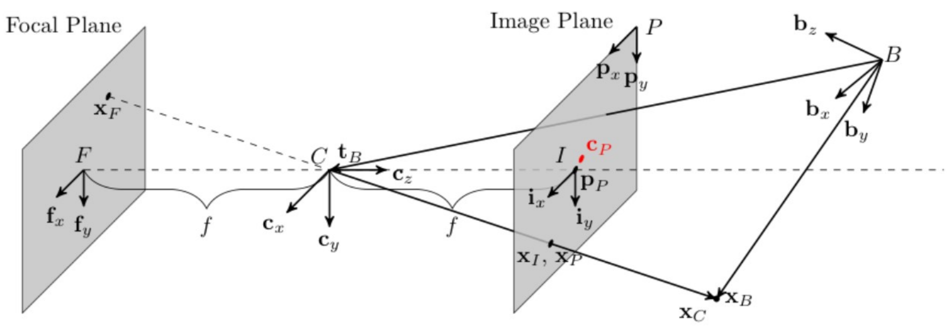


Figure 4.5: Pinhole Camera Model Reference Frame

In Figure 4.7 it is possible to see the particles and the Field of View of the Camera plotted in Camera Reference Frame. The FOV is characterized by the angles 44° and 32°, for the horizontal and vertical direction respectively [48]. Instead, as maximum depth was chosen 3 km, representing the limit beyond which the particles cannot be detected anymore because of their faintness, this value was decided considering an orbiting distance from Bennu $1.6 - 2.1$ km in the mission phase when the particles were detected [34]. The particles resulting in actual detections on the camera sensor are only the ones in the camera field of view.

Transformation to Pixel Coordinates

Finally the particles position in pixels are computed as they were captured on an digital sensor. Following the approach shown in Section 2.2. The values of the intrinsic camera matrix were computed as

$$k_x = Horizontal\ Sensor\ Resolution/Sensor\ Width \tag{4.5}$$

$$k_y = Vertical\ Sensor\ Resolution/Sensor\ Height \tag{4.6}$$

$$p_x = (Horizontal\ Sensor\ Resolution - 1)/2 \tag{4.7}$$

$$p_y = (Vertical\ Sensor\ Resolution - 1)/2 \tag{4.8}$$

Where the values for the $Horizontal\ Sensor\ Resolution$, $Vertical\ Sensor\ Resolution$, $SensorWidth$ and $SensorHeight$, are taken from [6], and the results are approximated to the nearest integer. Resulting in camera parameters listed in Table 4.5.

Camera Parameter	**Value**
Sensor Width $[mm]$	5.7
Sensor Height $[mm]$	4.28
Horizontal Sensor Resolution $[pixel]$	2592
Vertical Sensor Resolution $[pixel]$	1944
k_x $[pixel/mm]$	454
k_y $[pixel/mm]$	454
p_x $[pixel]$	1295
p_y $[pixel]$	971

Table 4.2: Camera Parameters [6]

Finally the intrisic camera matrix is calibrated as, performing a transformation $R^3 \Rightarrow R^2$

as shown in **??**, following the steps of [35] as used in [36].

$$\mathbf{K} = \begin{bmatrix} k_x & 0 & p_x \\ 0 & k_y & p_y \end{bmatrix} \tag{4.9}$$

4.3. A Test of Automated Particle Tracking Algorithm

Prior to the creation of a new automated tracking algorithm, previous functions of the nearest models needs to be adequately tested and assested, prior to their usage in critical segments of the current implementation. The tested algorithm is described in Section 3.3. Two key functions in particular are being examined and assested.
The first one being the inverse projection model used to obtain the coordinates in 3D inertial reference frame of the particles from their pixel coordinates of a digital sensor, which has been depicted in Section 2.3. In the considered framework, the implemented map, is employed in order to assemble, from the set containing the detections produced by all the particles in a determinate image, the subset of the nearest fragments to a considered detection in the prior images. This operation is performed, provided the x and y pixel coordinates, once the line of sight vector is extrapolated as in 2.3. Then, it is assumed that the distance of the particles from the spacecraft is equal to the distance of the probe from the main body. This will be called *Hypothesis 1* from here on. The particle inertial position being derived from this assumption.
Once the particle inertial position has been computed, the particle is assumed to be stationary, *Hypothesis 2*, and its relative position with respect to the spacecraft is computed in the subsequent frames captured up to an hour from the first image.
In order to test the validity of the assumptions, a testing environment as described in 4 has been adopted. In the simulated environment a number of particle have been generated and propagated, resulting in a varying number of measurements to be converted. The described simulation considers the classes of ejected particles and orbiting particles singularly as well as together, moreover a number of simulation for each one of the cases has been carried out, in order in account for different distribution in space of the particles, caused for example by different ejecta cone angles and prospect from the spacecraft.

4.3.1. Testing the validity of *Hypothesis 1*

In the given context the employed error metrics are computed as following.
Given a set of n detections, the position error vector for the detection k is

$$\vec{e}_k = \vec{r}_{t,Est} - \vec{r}_{t,True} \tag{4.10}$$

Where $\vec{r}_{k,Est}$ and $\vec{r}_{k,True}$ are respectively the predicted and the true position at time t, and $\vec{e}_k$, $\vec{r}_{k,Est}$, $\vec{r}_{k,True} \in R^3$ The first two metrics are necessary to establish whether or not is present a direction where the algorithm is more prone to error, or if the error is spread out across all directions.

- The average error along each coordinate, $\vec{\mu} \in R^3$

$$\vec{\mu} = \begin{bmatrix} \mu_x & \mu_y & \mu_z \end{bmatrix} = \frac{1}{n}\sum_{k=1}^{n} \vec{e}_k \tag{4.11}$$

- The covariance matrix, 3x3, of the position error

$$\mathbf{\Sigma} = \frac{1}{n}\sum_{k=1}^{n}(\vec{e}_k - \vec{\mu})(\vec{e}_k - \vec{\mu})^T \tag{4.12}$$

The second couple of metrics instead is deputed to evaluate the absolute magnitude of the error itself, allowing for a direct comparison with the scale of magnitude of the system

- The mean of the norm of position error, $\mu \in R^1$

$$\mu = \frac{1}{n}\sum_{k=1}^{n} ||\vec{e}_k|| \tag{4.13}$$

- The variance of the error norm

$$\sigma^2 = \frac{1}{n}\sum_{k=1}^{n}(||\vec{e}_k|| - \mu)^2 \tag{4.14}$$

Moreover, to give a reference to the error magnitude the mean diameter of the asteroid is 490 m, while the orbit of the OSIRIS-Rex Mission, when observing the ejecta events, ranged between 1.6 and 2.1 km.

Simulations Performed	100
Particles Generated in each simulation	50
n $[-]$	3547
$\vec{\mu}$ $[km]$	$\begin{bmatrix}0.23880 & 0.19928 & 0.11462\end{bmatrix}$
μ $[km]$	0.34356
$\mathbf{\Sigma}$ $[km]$	$\begin{bmatrix}0.03306 & 0.02877 & 0.01346 \\ 0.02877 & 0.03519 & 0.01756 \\ 0.01346 & 0.01756 & 0.01561\end{bmatrix}$
σ^2 $[km]$	0.07571

Table 4.3: Error Metrics for Ejected Particles

Simulations Performed	100
Particles Generated in each simulation	150
n $[-]$	1394
$\vec{\mu}$ $[km]$	$\begin{bmatrix}0.35429 & 0.22828 & 0.15286\end{bmatrix}$
μ $[km]$	0.46528
$\mathbf{\Sigma}$ $[km]$	$\begin{bmatrix}0.05377 & 0.02470 & 0.02062 \\ 0.02470 & 0.02188 & 0.01077 \\ 0.02062 & 0.01077 & 0.02245\end{bmatrix}$
σ^2 $[km]$	0.08261

Table 4.4: Error Metrics for Orbiting Particles

Ejected Particles

In order for the results to have statistical relevance 100 simulations are performed. In each simulation 50 particles are generated. The number of particles present in the field of view of the camera varies from simulations to simulation because the particles are generated in a random way as described in 4.2.1.
Following, the error metrics are computed as described above. Specifically n is the sum of the number of detections across all the simulations. This is, the sum of the number of performed conversions from pixel to inertial coordinates in each simulation across all simulations. The results are listed in 4.3.

Orbiting Particles

In this case a greater number of particles particles was generated, accounting for their increased sparsity with respect to the ejected case. The same number of simulations Similar results in terms of accuracy can be found from simulation to simulation.

The results are listed in Table 4.4.

Results Overview

Simulations considering other starting conditions, for example a mixture of ejected and orbiting particles, did not revealed any further trend to be highlighted. In all the performed simulations the accuracy achieved wasn't high, opening space for a debate about the applicability of the algorithm in a real world scenario.
In particular, a position error between 300 and 500 m cannot be considered acceptable in a system where the diameter of the main body is roughly 490 m. Moreover, the analysis showed that the position error committed is not made up of the z component only, but it involves all the three directions, highlighting that the algorithm cannot be applied even for a restriction, estimating the position of the object only for the x and y directions.
Moreover, particular orbital cases, were the obtained accuracy was high enough to justify its usage, leading to a possible, even tough circumscribed application, were not found either.
The goal of the current investigation is to find algorithms having decent properties of scalability, meaning that the result of their usage depends, as small as possible, from the application they are employed in, despite they produced adequate results in a particular use case.

4.3.2. Testing the validity of *Hypothesis 2*

The second investigated assumption is the static nature of the particles. In contrast what has been observed before, this case it is expected to show a difference in the error metrics, because of the difference in velocity between the particles ejected from the surface and the ones orbiting the asteroid. Resulting, in the second case, in a smaller difference of the velocity when compared to the assumed speed.

Error Metrics

To investigate the correctness of this assumption the error of the detection e_k is measured as the norm between the pixel locations of the expected position $\vec{x}_{Pix,Est}$ and the true position $\vec{x}_{Pix,True}$.

$$e_k = ||\vec{x}_{Pix,Est} - \vec{x}_{Pix,True}|| \tag{4.15}$$

With $\vec{x}_{Pix,Est}$ and $\vec{x}_{Pix,True} \in R^2$. Then for each image index after the reference, the error is averaged for all the detections in that image

	Ejected	Orbiting
Simulations Performed	100	100
Particles Generated in each simulation	100	150
n [−]	3547	1394
Shot Id	$\bar{e}_k$ **Ejected** $[pixel]$	$\bar{e}_k$ **Orbiting** $[pixel]$
0	0.0	0.0
1	159.6784	61.7585
2	387.7937	155.5397
3	534.2166	202.4903
4	790.0371	311.51156
5	926.8918	318.7804
6	1132.8800	378.9433
7	1294.5845	421.11198
8	1446.7523	531.5336
9	1467.2598	576.9501

Table 4.5: Average error norm by image

$$\bar{e}_k = \frac{1}{n} \sum_{i=1}^{n} e_k \tag{4.16}$$

The number of simulations and particle generated of the testing *Hypothesis 1* holds true also for the testing of *Hypothesis 2*. Also the parameter n is defined as before.

Results Overview

Graphs 4.8 show the evolution of the error magnitude in time. The difference in error obtained is significant, yet Subfigure 4.8a highlights that, even in the best conditions, the algorithm is committing an error not to be underestimated. In Table 4.5, the errors made in two cases are shown in greater detail. From these results emerges that also this part of the algorithm has limits in its employability. In fact, considering that the resolution of the sensor is 2592×1944 pixels, accounting for an error reached after a few iteration, 400 pixels for example, the algorithm is committing an error equal to around one fifth of the overall camera field of view.

4.3.3. Summary

In this chapter the simulating methodologies are detailed, moreover the measurement pipeline has been presented, finally an evaluation of the precision of past methodologies has been carried out. It is important to notice that the two tested algorithms are not used directly to compute the states of the particles of the ejecta events, instead acting as guidelines of which potential particles can be linked to other detections. Nevertheless their performances and the estimated errors obtained, make their capabilities not robust enough for their further application. Moreover, it has to be accounted that they are applied concurrently in the algorithm pipeline, leading to a sum of the respective committed errors in the final result.

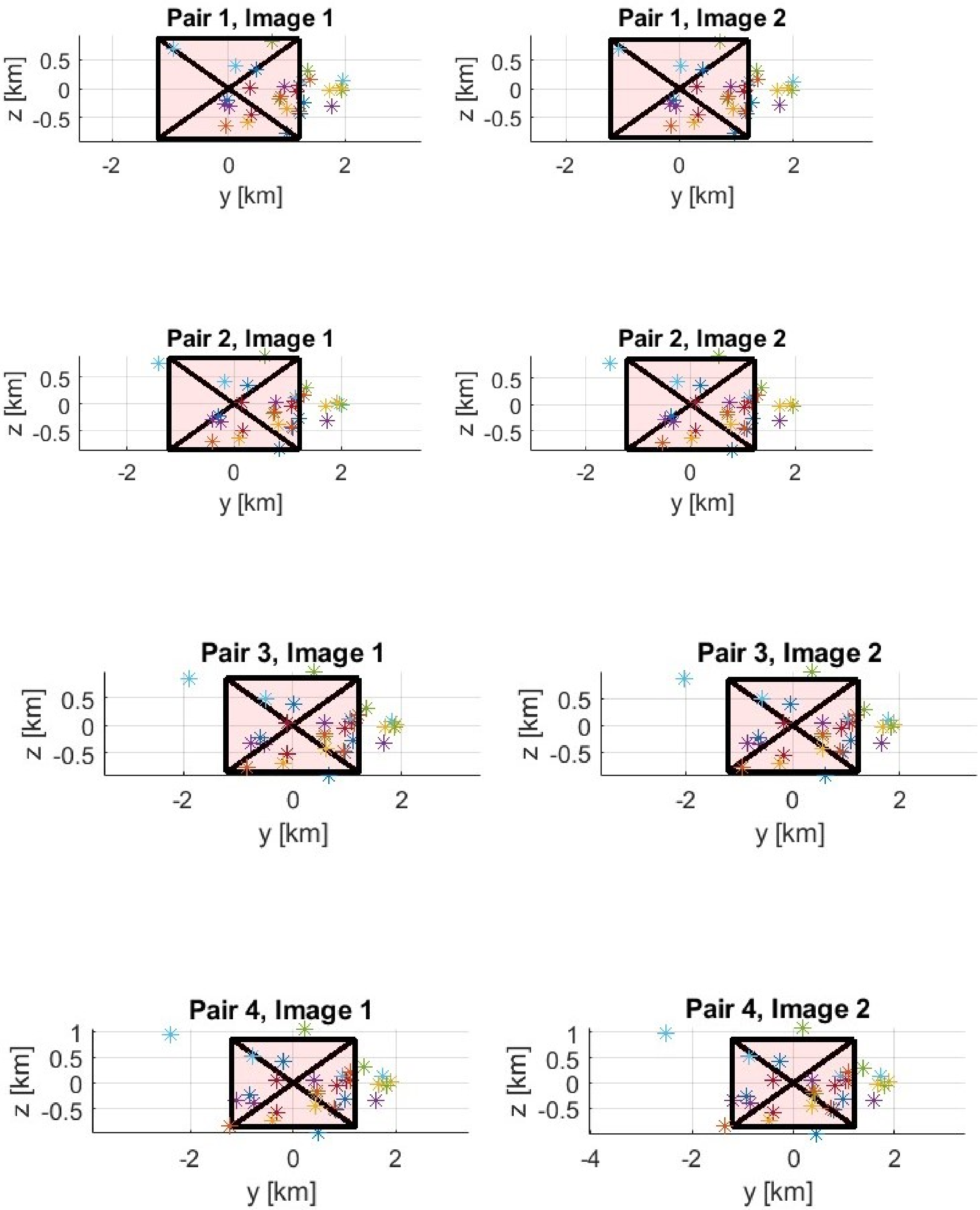


Figure 4.6: Particles and camera FOV, as seen in camera frame, based on the shooting rate of OSIRIS-REx, characterized by two photos taken with a short interval between each other every twenty minutes, the photos are shown in couples.

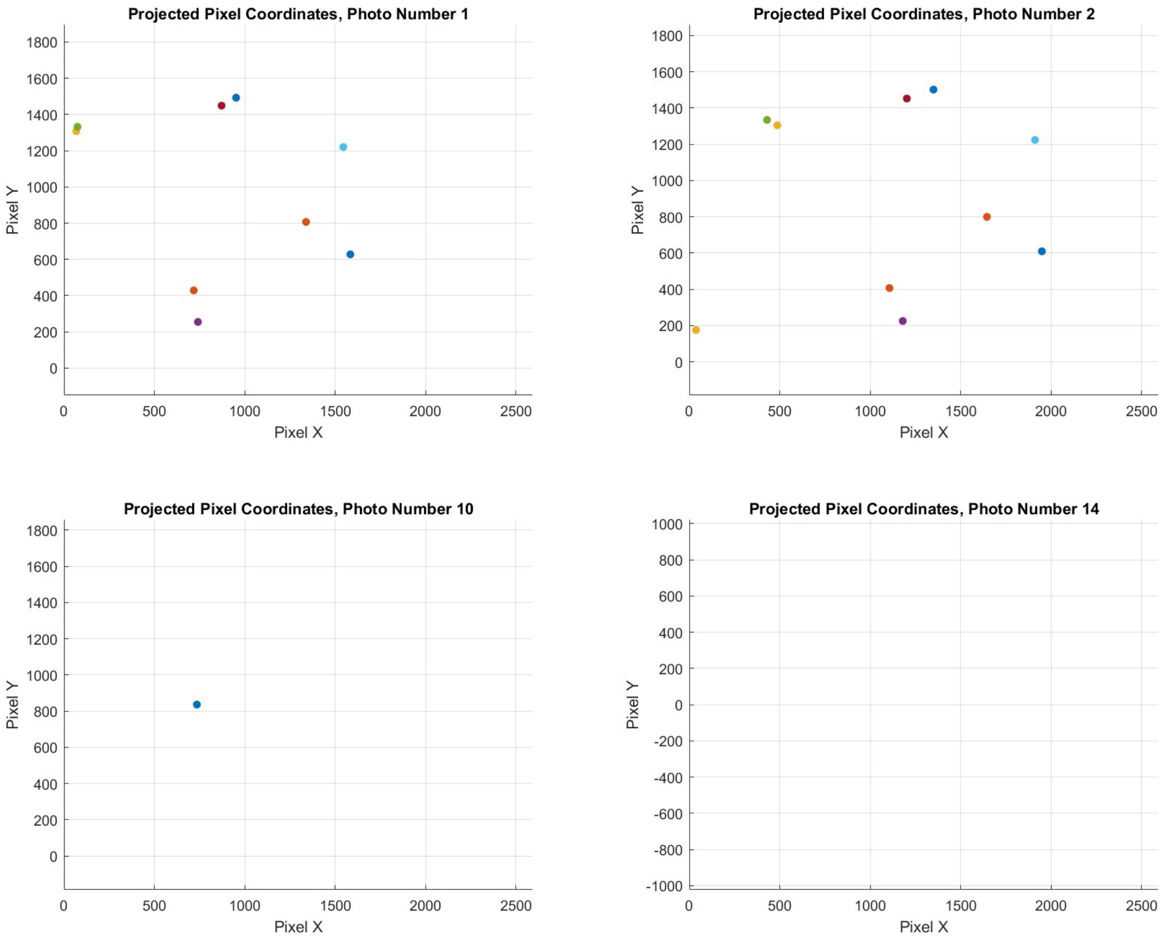
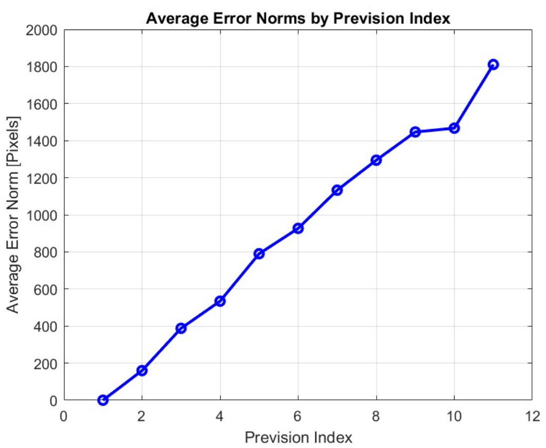
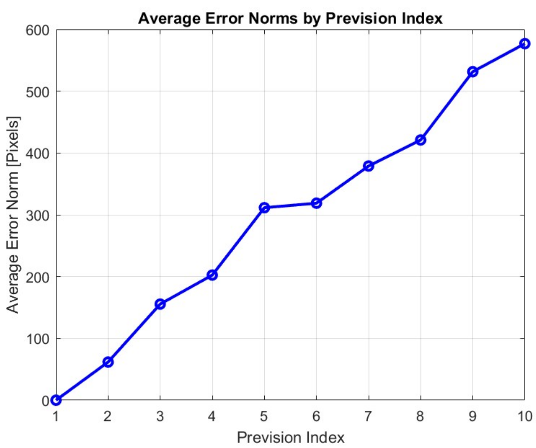


Figure 4.7: Particle in plotted in pixel coordinates on a grid of 2592x1944 pixels



(a) Particles ejected from the surface



(b) Particles orbiting the asteroid

Figure 4.8: Evolution in time of the error norm

5 | New Algorithm Development

5.1. Introduction

Because of the highlighted drawbacks in the approach produced by the literature, an alternative design of an automated particle tracking algorithm was deemed necessary. The new algorithm, although operating with a similar structure to the one already presented one, has been stripped of the all the parts capable of granting low levels of accuracy, and in particular

- The hypothesis that the distance of the particles from the spacecraft is equal to the distance of the probe from the main body, Section 4.3.1.
- The hypothesis that the particle is stationary, Section 4.3.2.

Moreover, the past algorithm, has been revisited in other keys aspects of its implementation, such as the filtering tool employed.

5.2. Kalman Filters Comparison

The state estimation problem for nonlinear stochastic dynamic systems can be solved with a variety of means. As described in 2.4 many possible alternative implementations of the filter originally introduced by Kalman are possible. The designed choice of the initial particle tracking algorithm filter fell back on the Extended Kalman Filter, EKF, this choice is here being questioned, to find out if an alternative solution to be implemented exists. Despite not being the main source of error of the previous algorithm, more reliable solutions exits and are being employed nowadays. The usage of batch filters is limited by the need of an online measurement processing, subsequently, assuming the requirement of non linear estimation, possible classes of filters that can be considered in substituting the EKF are the following ones:

- The Unscented Kalman Filter
- Particle Filters

- The Gaussian Mixture Probability Hypothesis Density Filter

UKF vs EKF

A comparison of the two platforms based on angles only spacecraft localization for Earth-Moon transfer orbits can be found in [16]. In this study the UKF proved itself more capable of the EKF regarding average localization accuracy and consistency of the estimates.

Particle Filters

Starting from the covariance ellipsoid of each particle, it is possible to run a Monte Carlo simulation generating an high number of particles within this volume. As [15] highlights this is one of the most accurate ways to solve the estimation problem, but the required computational load was the highest between the investigated filters.

5.2.1. The Gaussian Mixture Probability Hypothesis Density Filter, GMPHDF

While the previous approaches allocate a single filter for each potential followed track, cloning the filter when multiple associations are feasible, the algorithm propagates recursively in time the posterior intensity, which is first-order statistic of the random finite set of targets [62]. While very promising, this algorithm lacks of real world backing examples and applications in the real world orbit determination scenarios. Moreover tracking ejecta near asteroids with a standard Gaussian Mixture PHD filter faces three key limitations:

- Fixed detection and survival probabilities, the GM-PHD assumes constant, state-independent p_D and p_S, which breaks down when particle visibility fluctuates with geometry, illumination, and shadowing near a small body [63].
- Poor resolution of closely-spaced targets, overlapping intensity peaks lead the filter to merge or lose tracks of clustered ejecta, making it difficult to maintain distinct trajectories in dense swarms [14].
- Lack of physics-based integration, its reliance on simple linear–Gaussian motion models precludes embedding domain-specific dynamics—such as irregular gravity fields, solar radiation pressure (with eclipses), or ejecta velocity distributions—limiting accuracy in the complex environment around an asteroid [41, 59].

5.2.2. Final Choice

Considering the aforementioned aspects the final choice of the EKF replacement is represented by the UKF. The motivations driving this choice were:

- Its superior performances in accuracy and convergence velocity when compared to the EKF.
- The computational load of particle filters, making their usage not feasible in the current setting, requiring the tracking of a high number of targets at the same time.
- The lack of real world applications, fixed detection and survival probabilities, poor resolution of closely-spaced targets, lack of physics-based integration by the GM-PHDF

5.3. Tree Graphs

Tree graphs are a type of data structure, their employment in the automated particle tracking algorithm makes necessary a brief introduction followed by how they were applied in algorithm.
Firstly, three graph are composed by nodes and edges. The nodes or vertices are the set of fundamental units or points present in the graph, each node can represent an entity, an object or a state. The nodes, instead are connections between pair of nodes, each edge connects exactly two distinct nodes. In the present study the tree graph is a directed tree graph, meaning that the each edge between two nodes has a start and an end node, and information can only flow following this precise order. In MATLAB a tree graph with directed edges is initialized by the function *digraph*.
It the current settlement each nodes represent a single detection of a single particle, the data linked to that detection is encapsulated in its precise node. The data contained in a single nodes varies in time, depending on which phase the algorithm is in, and which are the quantities computed up to that time. In general each node will be associated to the following quantities

- Node Number, this is solely a node identification number, not carrying information that will be actually used in the tracking of the particles. Noticeably, as it will be described later on, the same node will have different identification number during the data elaboration.
- Ephemeris Shot Time, in seconds.
- State, this is the estimated 3D state, position and velocity, of the particle for that

precise x and y pixels measurement.

$$\begin{bmatrix} x & y & z & v_x & v_y & v_z \end{bmatrix} \tag{5.1}$$

- Covariance matrix associated to the estimated state as a 6×6 matrix.

$$\mathbf{P} = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{xz} & \sigma_{xv_x} & \sigma_{xv_y} & \sigma_{xv_z} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{yz} & \sigma_{yv_x} & \sigma_{yv_y} & \sigma_{yv_z} \\ \sigma_{xz} & \sigma_{yz} & \sigma_z^2 & \sigma_{zv_x} & \sigma_{zv_y} & \sigma_{zv_z} \\ \sigma_{xv_x} & \sigma_{yv_x} & \sigma_{zv_x} & \sigma_{v_x}^2 & \sigma_{v_xv_y} & \sigma_{v_xv_z} \\ \sigma_{xv_y} & \sigma_{yv_y} & \sigma_{zv_y} & \sigma_{v_xv_y} & \sigma_{v_y}^2 & \sigma_{v_yv_z} \\ \sigma_{xv_z} & \sigma_{yv_z} & \sigma_{zv_z} & \sigma_{v_xv_z} & \sigma_{v_yv_z} & \sigma_{v_z}^2 \end{bmatrix} \tag{5.2}$$

The vertices instead are associated to a starting and an ending node, along with an optional weight. In the study, the vertices represent possible connections between nodes appertaining to the same sequence, the weight is present and computed as the reciprocal of the euclidean distance in pixels between starting and ending node.
Starting from the basic information available, edges and numbers contained in the vertices are computed and refined, creating the final association map, which is composed by an ordered sequences of measurements.

5.4. Portrayal of the Automated Particle Tracking Algorithm

5.4.1. Initial Nodes Allocation

At the very start, the only available information available are the measurements in pixel coordinates, i.e. pixel x and pixel y. Accordingly, a set containing an equal number of nodes is initialized, each node of the algorithm containing only the two pixel coordinates of one measurement. Concurrently, the fields of state and covariance are initialized with an initial state and covariance which are respectively composed solely by zeroes [1]. Lastly, for each node, the ephemeris shot time is filled with the shooting time of the measurement.

[1]Due to MATLAB's syntax requirements, these fields could not be left blank.

5.4.2. Measurements Handling and Sequencing

Once the set containing all the nodes is initialized and added to the graph, in order to establish the various edges, the possible connections between nodes are identified. In this pipeline, the nodes are processed sequentially, accounting for the order of the image they belong. The linkage process starts from the measurements of the first image, in particular, each pixel location in the first available shot is considered singularly, looking for possible connectable detections in the subsequent images. While the order in which the images, and the detections in the images remain the same as the previous algorithm, the way the detections are analyzed and connected, as well as the branching logic, differs from the previous implementations. Once the chain of the linkable measurements starting from the initial detection has been completed, either because there are no more measurements respecting the linkability properties or the images are finished, the sequence, usually having a ramified structure, undergoes a backward smoothing process. The result of the backward smoothing is a pruned sequence, i.e. a clean track with no ramifications, describing the final chain of measurements, that will be later saved in the set of the identified tracks. Moreover, the detections in the pruned sequence will be eliminated from the set of the detections, thus will not be considered for subsequent sequence identifications. The same process occurs for each detection of the first image, when all the detections in the first image have been investigated, the same process is carried out on the detections of the second image and so on, until every image in the sequence has been analyzed.

5.4.3. Measurement Linkage and Sequence Identification

Whilst the process aforementioned process is performed, as more and more x and y detections are ingested by the Kalman filter computations are occurring. In particular, selected a singular detection on a measurement level, i.e. considering the x and y coordinates, a search for linkable connections in the subsequent image begin. A first search in an area of 500 pixels from the considered detection is carried out, computing the euclidean distance between couples of measurements. The value of 500 pixels was chosen empirically, via a trial and error process, and it is strongly dependent from the properties of the studied ejection event and the employed image cadence, along with the physical magnitude of certain orbiting properties of the main body of the system. Simulating ejection events as the ones described for the case of Bennu it was found that a smaller area frequently missed in associating detections.

This choice already marks a difference with past employed algorithms, that applying the following process

- Selecting a pixel measurements
- Compute the corresponding inertial state
- Computing its position in the measurement space, x and y coordinates, applying directly the pinhole camera model
- Look for subsequent associations in the area centered in the aforementioned pixel coordinates with an adaptive distance threshold
- The area is linearly shrunk as more and more measurement arrive

The described process was introducing two possible sources of error. Following the assumption that the estimate of the Kalman Filter is getting closer and closer to the true solution.

Although elegant, this process is not robust against errors committed in the state estimate operated by the Kalman Filter, especially in its first iterations.

For this reason it was decided to center the area threshold for subsequent detections pairings at the measurement itself, instead of at x and y coordinates computed by applying the pinhole camera projection to the estimated state coming from the Kalman Filter.

Subsequently, the area threshold is employed to link the detections with the order explained in 5.4.2.

While the possible connectable nodes are computed, the kalman filter propagation process proceeds accordingly. Every time that a pair of linkable detection is found, the corresponding measurements coordinates are fed into the UKF, outputting the state of the new ingested measurement and the associate covariance.

5.4.4. Unscented Kalman Filter

The employed UKF resembles closely the strucure already described in 2.4.3. The used tuning parameters are listed in 5.1.

Param	Value
σ $[pix]$	$\begin{bmatrix} 10.5^2 & 0 \\ 0 & 10.5^2 \end{bmatrix}$
α	10^{-3}
P_0	$\begin{bmatrix} 5.6\times10^{-2} & 3.5\times10^{-2} & -7.1\times10^{-2} & 0 & 0 & 0 \\ 3.5\times10^{-2} & 9.7\times10^{-2} & 7.6\times10^{-2} & 0 & 0 & 0 \\ -7.1\times10^{-2} & 7.6\times10^{-2} & 8.1\times10^{-2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.8\times10^{-9} & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.7\times10^{-9} & 0 \\ 0 & 0 & 0 & 0 & 0 & 9.6\times10^{-9} \end{bmatrix}$
Scaling Factor	10^{-2}

Table 5.1: Unscented Kalman Filter Tuning Parameters

Where σ is the noise matrix, α the spread of the sigma points, P_0 the first Guess Covariance, and the scaling factor is a multiplicative factor applied to the covariance of the first guess. Concerning the dynamics instead, the filter models the gravitational acceleration only, exploiting a keplerian model.

$$\vec{a} = -\frac{\mu}{r^3}\vec{r} \tag{5.3}$$

With μ as listed in 1.2, and $\vec{r}$ is the distance in inertial coordinates.
The measurement function instead, resembles the one already described for the simulation of the image acquisition pipeline in 4.2.2. Finally, the filter is initialized using as position, the coordinates produced by the inverse projection of the pinhole camera model, as described in 2.3. The initial velocity was instead set to zero.

5.4.5. Linkage Conflict Handling

Possible conflicts can occur when two edges, coming from two branches of a ramified sequence are converging on the same measurement. At this stage, before the pruning process, the problem is not the one of deciding which one of the two connections in the right one to be made, subsequently deciding the correct sequence of measurements. In fact, it has to be highlighted that both the possible connections are retained at this stage, with the problem being which measurements to use in order to propagate the Kalman Filter up to the measurement where the two branches converge, in order to avoid using more complex logic for handling the estimates.

Unfortunately, the natural choice of using the covariances predicted by the UKF, subsequently selecting the allegedly correct connection as the one minimizing the covariance of the estimate, proved not to be enough accurate for the first estimates, produced with a small number of ingested measurements. In contrast, selecting the edge with the highest weight, i.e. the smaller distance, was found to be enough accurate for the vast majority of the cases, although the designated metric, creating limits to the generalization capabilities of the algorithm, thus making it not robust to the algorithm adoption in different frameworks, cannot be considered the ultimate design choice and more in depth studies need to be carried out on this subject.

5.4.6. Sequence Pruning Process

A sequence ends when the images are finished or when the remaining detections do not meet the requirements to be connected. Then, the found branched detections tree has to be pruned in order to resemble the final result of the automated particle tracking.

In order to to this the following methodology is employed. First, the nodes having only one connection are identified, these class in then divided into start and end nodes. Doing so, all the possible starts and finish nodes of all the possible sequences, being part of a main ramified tree graph, are found.

Consequently, for each sequence, its total covariance, computed as the sum of all the covariances associated to the estimated states of each node in that sequence is computed. Of this 6x6 matrix is computed the trace, and then each trace is normalized for the number of detections composing that sequence. Doing so, a sort of average covariance for estimate is computed. The creation of this metric was necessary to be able to compare the covariances of sequences having different number of measurements.

Nevertheless, it is worth mentioning that in this way the estimate best sequence, the one with the lower covariance for estimate, is inevitably driven towards longer sequences. This aspect however, complies with the real world, where the true track is usually one of the longest. This aspect originates because usually a wrong branch of the three runs out of connectable measurements in a short time, producing a sequence with a lower number of detections connected.

It has also to be highlighted that the hardest case for this pruning strategy occurs when two sequences have an almost equal number of measurements, with their length varying of ± 1 measurement, in this case decide the correct ending node may prove to be tricky, causing the hightest amount of association errors within a correctly identified sequence.

5.4.7. Resetting the Data Structure

Once a sequence ends, according to the criteria described above, the nodes, along with their pertinent relevant quantities are extracted from the dataset and stored. Prior to the selection of the next measurement marking the start of a new sequence, and so another sequence individualizing cycle, the graph is cleaned of all the information used in the prior cycle. The nodes composing the final sequence are erased from the dataset initialized as 5.4.1. The states and covariances of the remaining nodes are brought back to their initial states of only zero terms, the edges between vertices not part of the identified sequence are removed from the graph.

5.5. Performance Evaluation

The two performance metrics employed for the evaluation of the performances of the algorithm are now introduced. Later on, the results obtained by the algorithm will be shown.

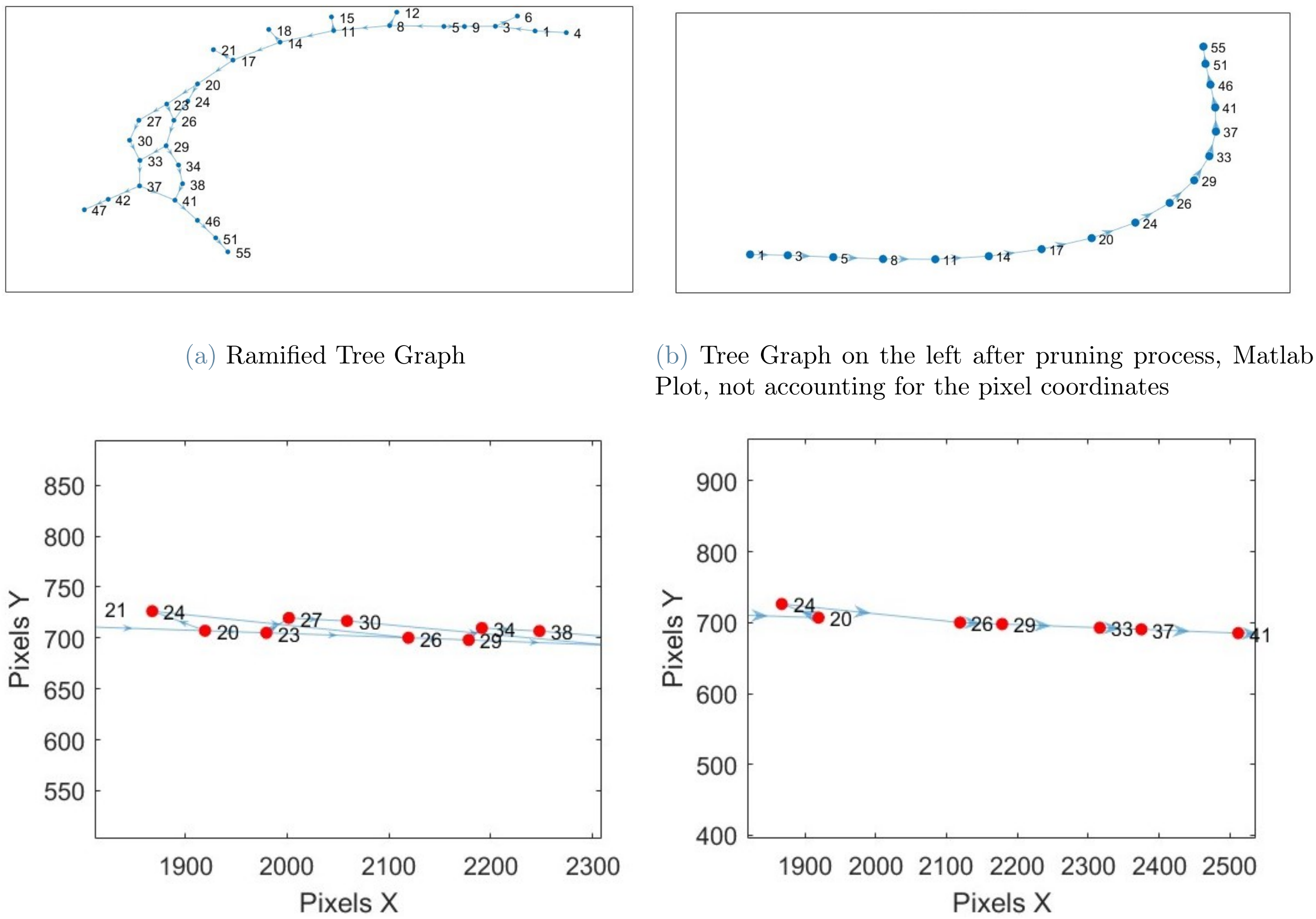
5.5.1. Error Metrics Definition

The automated target tracking algorithm has two main goals performed concurrently. The first is the correct sequence identification, so being able to connect correctly all the detection effectively produced by the same particle from frame to frame, known also as tracking. The second task performed by the algorithm is to correctly predict the state of the particle when acquiring a sequence.

Sequence Identification Metric

In order to evaluate the capability of the algorithm in acquiring the correct sequence, for each sequence, in each simulation, the following parameters are computed.

1. Starting from the estimated sequence, the corresponding correct true sequence, between all the known, computed true sequences. Representing the evaluation the pairing capabilities of the algorithm on an high level. This index is computed com-



(a) Ramified Tree Graph

(b) Tree Graph on the left after pruning process, Matlab Plot, not accounting for the pixel coordinates

(c) Ramified Tree Graph, Pixel Coordinates, Zoom

(d) Pruned Tree Graph, Pixel Coordinates, Zoom

Figure 5.1: Tree Graph Processing
Noticeably, an error has been committed in the identified track, with the right connection to be made moving on from the node 20 being 23 instead of 24

paring the estimated sequence with the known present sequences, and finding the most similar one, the one with the highest number of equal measurements, x and y coordinates at equal ephemeris shot times.

2. The length of the estimated identified sequence, so the number of measurements paired by the algorithm.

3. The length of the true sequence, the number of measurements composing the true sequence, identified as in 1.

4. The correct pairs, computed as the correct associations, same x and y coordinates, same ephemeris shot times, between the estimated and the true sequence.

As an example the performance evaluation for a trivial case is presented in 5.2. The

Id True Seq.	Length True Seq.	Length Est. Seq.	Error Type	Correct Pairs
4	15	17	More	14
5	17	14	Less	13
17	25	25	Equal	25
13	17	17	Equal	17
18	18	18	Equal	18

Table 5.2: Unscented Kalman Filter Tuning Parameters

case is trivial in the sense that the low number, ranging between 3 and 4, of detections concurrently present in the field of view of the camera makes the task easy to be performed, later on more complex cases will be presented.

In this case the total number, the sum of all the detections in all the photos, or the sum of the true lengths of the true sequences, was 93. The number of correctly paired measurement was 88, with a success ratio of 94.565%. The difference between 93 and 88 is 5, comprising the miss in

- Connecting a particle to the first sequence, 14 the correct pairs vs 15 the true length
- 4 missed pairing to the second sequence, 13 the correct pairs vs 17 the true length

Moreover, two particles were not paired to any sequence. If the 3 incorrectly paired pair of the first sequence, $17 - 14 = 3$ (x), go to the second sequence, and the first and second sequence takes one unpaired particle each, all the lengths and the total number of the particles are respected, with the first one being

$$Correct\ Pairs\ +\ One\ Unpaired\ Particle\ =\ True\ Length \tag{5.4}$$

$$14 + 1 = 15 \tag{5.5}$$

and the second

$$Correct\ Pairs\ +\ One\ Unpaired\ Particle\ +\ x\ =\ True\ Length \tag{5.6}$$

$$13 + 1 + 3 = 17 \tag{5.7}$$

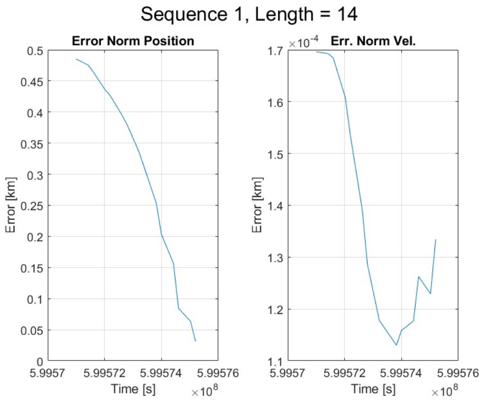
State Estimation Metric

The second goal of the automated target tracking algorithm is to estimate the state of the particle for each fed measurement of each found sequence. In this regard, in order to evaluate the achieved performances, the norm of the state error, divided into position and velocity, is computed. The state error for position and velocity are respectively defined as

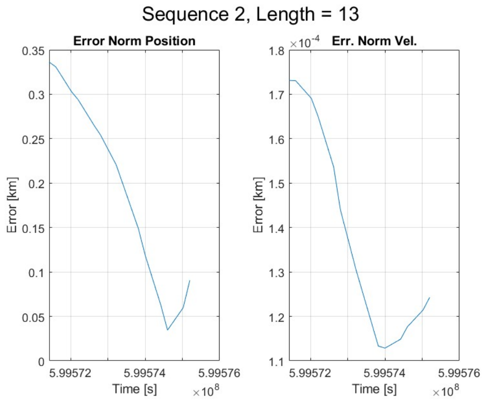
$$e_r = ||\vec{r}_{est} - \vec{r}_{true}|| \tag{5.8}$$

$$e_v = ||\vec{v}_{est} - \vec{v}_{true}|| \tag{5.9}$$

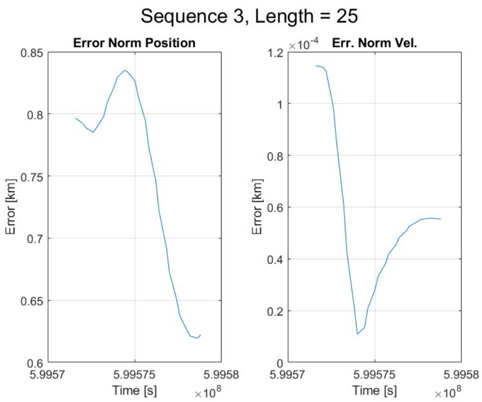
Where $\vec{r}_{true}$ and $\vec{v}_{true}$ are coming from the numerical integration of the particles state and $\vec{r}_{est}$ and $\vec{v}_{est}$ are the ones estimated by the UKF.
The computation of the error is performed in the time instants shared by both sequences, thus, in case the estimated sequence comprises detections not present in the true sequence the error will not computed for those positions. Nevertheless, the estimated time instants have not been stripped away from the estimated sequence, nor the estimated sequence, after being stripped from the wrong associations has been ingested by the UKF for a second time. So, in case a wrong association has been done, the corresponding measurement has been accounted for in computing the sequence of corresponding state estimates, influencing this metric because it influences the states computed afterwards. Results coming from the aforementioned simulation are shown in 5.2.



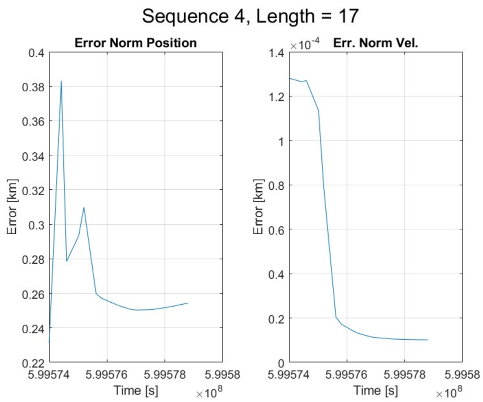
(a) Sequence 1



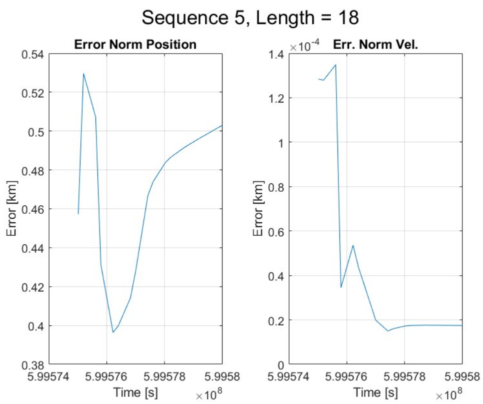
(b) Sequence 2



(c) Sequence 3



(d) Sequence 4



(e) Sequence 5

Figure 5.2: Estimated State Error: Position and Velocity.

It is worth noticing that the fifth sequence, despite having received the correct measurement, still shows diverging error behavior.

5.6. Testing

5.6.1. Experiment Set Up and Error Metrics

In order to adequately test the capability of the pipeline, the algorithm is tested with an increasing number of generated particles. It is important to note that this increase does not directly impact the system's performance, which is primarily determined by the number of targets simultaneously present in a single image. To ensure that the increase in generated particles will correspond to a raise in the target concurrently present in the field of view of the camera, it was decided to test the pipeline mostly with particles ejected from the surface of Bennu. The ability to influence the ejection direction by pointing the trajectories toward the spacecraft's future path, is used only to make sure they can be detected, with little impact on the actual outcome of the experiment. In contrast, orbiting particles are way harder to be created, maintaining the random generation requirements, in a specific region of space. Further on, particles ejected from the surface have usually higher velocities with respect the orbiting ones, placing the pipeline in the harshest environment possible. In fact, as described in 3.2, the algorithm inspiring this work has as scope the detection of the slower orbiting particles, while its counterpart, leveraging a linear motion assumption, is more suited for faster ejection events.
Following what has been said, the used metric to evaluate the performances of the algorithm will be the number of targets present in the images, considering that this quantity varies in time, for each experiment the average number of target will be computed along with the obtained pairing accuracy.
For example, starting from the trivial experiment presented before, generating 20 particles, for an average number of targets of 3, the obtained accuracy was 94.565%.

5.6.2. Monte Carlo Simulation

A Monte Carlo Simulation is mandatory for a complete testing of the algorithm. In this framework, an increasingly higher number of particles is generated. This will cause an increase in the average number of particles present at the same time on the sensor.
In order to have statistical meaning, for each quantity of particles the experiment is carried over 10 times, the generated quantities were

$$\begin{bmatrix} 1 & 5 & 10 & \cdots & 50 & 55 & 60 \end{bmatrix} \tag{5.10}$$

For each quantity the experiment is carried out ten times. It has to be noticed that, being the particles generated randomly, the number of generated particles does not translate directly into the number of present detections concurrently present in the field of view. Thus, it is not possible to compute error statistics for each generated quantity of particles but only depict the general trend of the achieved performances.

From each experiment are extracted the following error metrics

- The average number of targets in the camera FOV
- The Sequence Identification Metric, defined as 5.5.1

5.6.3. Results

The results, for a generated number of particles between 1 and 60, are shown in image 5.3. The accuracy of the algorithm can be considered sufficiently high up to a number of 6 or 7 detections in the field of view of the camera. From these values on, the performances of the algorithm are compromised and show a sharp decrease.

It has to mentioned that, the modeling choice of generating the particles with a gaussian distribution, already defended in 4.2.1, is negatively influencing the algorithm performances. Choosing an uniform distribution instead, and generating an equal number of particles, the probability of having particles close to each other is lower. A normal distribution, generating the particles more frequently near to its mean value easily creates simulations were the tracks of multiple particles are close to each other. In contrast, using a uniform distribution the particles would be spread out more homogeneously on the sensor, bringing a less sharp decrease in sequence detection accuracy with the number of particles generated and extending the region of usage of the algorithm.

5.6.4. Real World Validation

It has to be highlighted that, despite the ejection events on Bennu are well document in the literature, their time span is not mentioned, thus the ejection cadence of the particles cannot be computed, consequently the number of detections concurrently present in the FOV remains an unknown.

In the Monte Carlo testing, the value of 60 was chosen as the upper limit because it represents the upper limit of usage of the algorithm, with the performances decreasing drastically for higher values.

Moreover this value is similar or higher than the vast majority of the ejection events, being close, as order of magnitude, to the major ejection events.

A direct comparison with the results of the previous automated tracking algorithm is not possible, because the metrics given in [36], shows only a comparison of the results between the automated and the manual inspection tracks, without further investigation with respect to figures of merit deemed important in the current study.
In the mentioned study, the only given accuracy metric was that the automated particle tracking algorithm was capable of correctly found only 47.8% of the total number of tracks. Nevertheless this number resembles all the found detections in all the studied ejection phenomena, not giving us any further information in the comparison between the two methodologies.

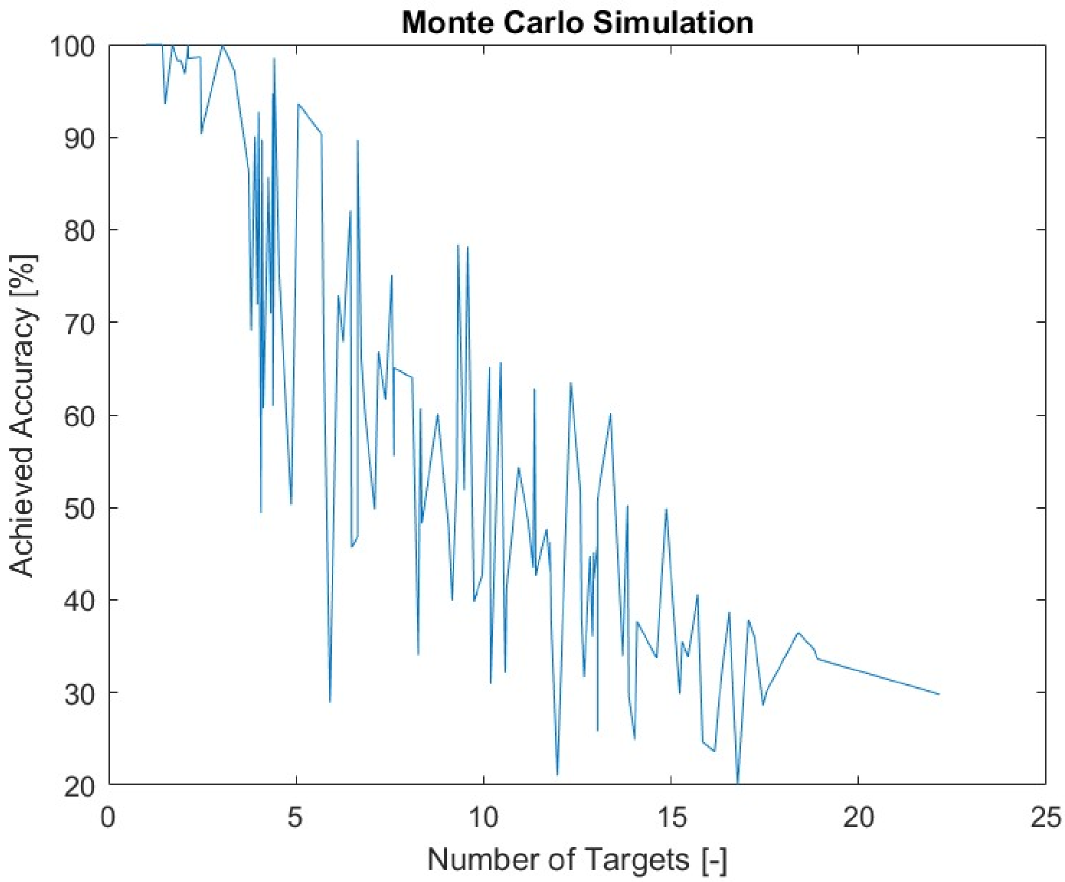


Figure 5.3: Monte Carlo Simulation Results

5.7. Summary

After having assested key subroutines employed in previous works, in this chapter are proposed novel advancements carried out in the current investigation. The performed testing highlighted a lack a precision in performing key operations with the original pipeline. Subsequently, the code was modified

- The extended Kalman filter was updated with the unscented Kalman filter
- The pieces of code leveraging the two hypothesis tested in the previous chapter, being not precise enough were stripped from the algorithm
- Finally the particle association logic was consequently revisited

Finally the obtained error metrics are shown, in particular are computed

- The obtained error in state estimation, highlighting how the UKF tends to converge ingesting more and more measurements
- The robustness of the algorithm to an high number of targets present concurrently in the FOV, granting decent performances until the number of targets in the FOV are less the ten

These metrics were carried out along with real world validation consideration and a comparison with the old automated particle tracking algorithm.

6 | Conclusions and Future Developments

6.1. Introduction

6.1.1. Starting Point

The starting point of this work is the automated particle association from frame to frame using the linear motion assumption. The linear motion assumption refers to the trajectory of the detections on the camera sensor, which follows a straight line, moreover the velocity, or pixel velocity using a sensor-wise notation is constant. This hypothesis are then feed into a linear Kalman filter for a linear estimation of the particle motion using the acquired measurements.

This is was the approach carried out by Lanza in [30], a similar approach, described in 3.2, and applied to the Bennu case study can be found in [47]. The described approach can be considered valid when the observer is placed far away from the observed object, and the last one is performing an orbit characterized by a semi major axis of important dimensions, preferable with a low eccentricity.

The described scenario doesn't hold true for the environments that can be found in small asteroids, when the non linearity of the particle motion becomes clearly visible. This effect occurs also because a lot of the particles orbiting small bodies are not able to perform a single revolution around the asteroid, resulting into particularly rounded shapes. Thus, the error committed using a linear motion assumption cannot be considered negligible anymore.

6.1.2. Objectives

The primal objective of this study was to develop an algorithm able to track a number of particles concurrently present in the camera field of view, creating a dense field of detections, without the employment of linear motion assumption.

The first task is the implementation of an adequate testing platform from scratch, its design need to be performed using the results obtained by the literature in the study of the ejection phenomena observed on Bennu. The second objective, or investigation task, consisted in a preliminary testing of various techniques already employed by the literature, evaluating their performances in sight of their potential application in a novel pipeline.
Finally, leveraging the results obtained in the two aforementioned stages, the study aims to formulate a novel algorithm.

6.1.3. Work Performed

The main work has started with the study of the environment and the equations used for its modeling, aiming to accurately simulate the ejection phenomena and correctly set the subsequent algorithm testing.
Further on, the main mathematical tools used in the investigation have been introduced. In particular, the most important filtering techniques and the algorithms employed to solve the automated particle tracking have been described in depth, along with the results obtained by the literature from the their implementation to the scope of the current research.
Lastly, thanks to an evaluation of the already employed techniques, the development of a new automated tracking algorithm has been completed. Subsequently, the obtained results were presented.

6.2. Results

6.2.1. Automated Tracking Algorithm Critiques

The first obtained results come from the testing of techniques already employed on the field that were based on the following hypotheses

- The distance of the particles from the spacecraft is equal to the distance of the probe to the main body, used in the computation of the particle state
- The stationary nature of the particles, used to link particle detection from one image to the others

For both the hypotheses their testing highlighted a low degree of accuracy, preventing their usage in further pipeline developments.
In particular, employing the first hypothesis, after having performed a number of environmental simulations, the average position error obtained ranged from 0.35 to 0.77 km,

to be compared with the mean asteroid diameter of 490 m.
The second hypothesis can create an error of almost one hundred pixels comparing two subsequent measurements. Moreover, the error increased rapidly in the following iterations, reaching a magnitude of eight hundred pixels in the worst conditions after nine iterations. As a term of reference the sensor resolution is 2592×1944 pixels.
On top of these considerations the two hypothesis are employed in the same algorithm concurrently, causing a sum of the produced errors in the final results.

6.2.2. New Developed Algorithm

Having evaluated the sources of inaccuracies present in the original automated particle tracking algorithm, it was decided to build a new pipeline, stripping the old one of the erroneous steps. The obtained code was then tested in its proficiency of link the detections generated by the same particle from frame to frame, and estimate the state from the used measurements. The obtained results were promising, but with potential growth opportunities.
The state estimation, performed by the unscented Kalman filter built into the particle tracking algorithm did not always performed as expected, moreover the number of detections concurrently in the field of view revealed to be the biggest bottleneck in the algorithm, causing a drop in the obtained accuracy.

6.2.3. Real World Validation

In order to achieve the desired results, in terms of accuracy and real world applicability, a consistent amount of time was spent in the study of the particle ejection phenomena itself and their simulation. The literature provided important results in this regard, comprehending in depth descriptions of the studied phenomena, but unfortunately one key information was missing. In fact, the description of the ejection phenomena, despite declaring how many particles were identified in each ejection phenomena, lacked expressing the time scale of the phenomena. Without any knowledge on how much time the ejection phenomena have lasted, the particle number ejected from the asteroid surface at the same time, thus present concurrently in the camera field of view, remains an unknown.
Without this piece of information, it is not possible to precisely know if the capabilities of the encoded pipeline are adequate to its employment in a real world scenario similar to the one encountered by the OSIRIS-Rex mission.

6.3. Future Developments

The current work has started from the lecture of a series of space-centered studies, detailing automated particle tracking algorithms developed for the space segment and, more particularly for the study of particles emitted by a certain celestial body, the asteroid Bennu.
These pipelines, despite having achieved promising results, primarily the capability of accounting for the non linear motion of the targets, were not developed accounting for other solutions of the same problem obtained in relatable research realms.
Considerably, the major bottleneck of the developed pipelines is considered to be the number of particles concurrently in the field of view of the camera, translating to detections closer and closer to each other. Notably, the current work do not grasp from any of the outcomes achieved by multiple target tracking algorithms developed for military or civil usage, for the specific task of associating the same detection from frame to frame.
Thus, considering the current work the very foundation of the study on how to track the particles revolving an asteroid, comprehending an accurate depiction of the already employed techniques, the future research should merge results obtained far from the initial space segment.

6.3.1. Source Research Fields

In recent years, to simultaneously track multiple, closely spaced, targets, the random finite set and the finite set statistics have been researched. From these investigations the probability hypothesis density (PHD) and the extended target PHD (ET-PHD) filters have been finally proposed. In this framework, measurement partitioning is of paramount importance [58]. It splits the full set of measurements into several non-empty groups and tries to find the partitions that are closest to the actual true ones. The performance of multiple extended target tracking depends a lot on how well the measurements are partitioned, since the result is directly used to update the predicted state of each target. To handle this issue, traditional partitioning methods usually look at all possible ways to divide the measurements, accounting also for the increased computational cost of handling more and more partitions.

6.3.2. The Measurement Partitioning Problem

In the years a number of partitioning algorithms have been produced. In the context of multiple closely spaced extended target tracking (MCETT), the targets may have parallel or crossed tracks, resulting in the following fundamental problems [58]

- How to maintain the accuracy of measurement partitioning in the presence of clutter and missed detections, without reducing the tracking performance?
- How to achieve high tracking performance in MCETT methods, especially when dealing with closely spaced extended targets in practical situations such as crossing or parallel tracks?

Distance Based Partitioning

Initially, a distance-based measurement partitioning algorithm, known as distance partitioning, was proposed. Leveraging the properties of the distances between measurements from the same target to reduce the number of partitions. This method wasn't without flaws, with the main drawbacks resulting in a quick increase of the computational cost as the number of targets grows. Also, the main assumption upon which the algorithm was based on, that measurements from the same target are close together, does not work well when targets are spatially close. In such situations, like when targets cross paths, all the measurements are grouped into the same area. Distance partitioning is likely to wrongly treat this group as coming from a single target, causing cardinality errors, mistakenly computing the number of targets during filtering [58].

Target Predictions Based Partitioning

In this methodology, building on the distance-based measurement partitioning algorithms, two more partitions are produced from the measurement set, the prediction partition and the EM partition. In the prediction partition, the predicted mean of a component is used as the center of the elliptical target.
The extension estimate $\hat{X}^{(j)}_{K+1|K}$, a symmetric positive definite matrix representing the elliptical shape of the j^{th} extended target, is used to group measurements.
Measurements that satisfy a distance condition with respect to the predicted mean and shape (within a threshold based on the inverse chi-squared distribution with 99% probability) are grouped into the same subset.

$$(z^{(i)}_k - m^{(j),d}_{k+1|k})^T (\hat{X}^{(j)}_{K+1|K})^{-1} (z^{(i)}_k - m^{(j),d}_{k+1|k}) < \Delta_d(p) \tag{6.1}$$

Where $m^{(j),d}_{k+1|k}$ represent the first d components of the predicted mean of j^{th} GIW component, $z^{(i)}_k$ is the i^{th} measurement at time k, $\Delta_d(p)$ is determined according to the inverse χ^2 distribution with d degree of freedom, and p is the probability and equal to 0.99. Sub-

sequently, the measurements not assigned to a subset are treated as individual subsets [17].
The main drawback of the described approach is to restrict the its application to targets of similar shapes and sizes, while it becomes difficult to track elliptical targets of different sizes.
To solve this problem, the EM partition method was introduced. It is based on the Expectation-Maximization algorithm for Gaussian mixtures and uses predicted components to initialize the mixtures. Both prediction and EM partition methods produce only one partition to be used in filtering.
Nevertheless, this still pipeline doesn't solve the case when the targets are close to each other.

Prediction-Driven Measurement Sub-Partitioning

In this last iteration, to effectively approximate all possible partitions, several reasonable partitions are considered to achieve stable and accurate target tracking performance. The partition number is determined accordingly to the possible changes of the target number. Suppose that the number of extended targets at time step $k-1$ is J_{k-1}, the number of newborn targets is $J_{\gamma,k}$, and the number of spawned targets from the same existing target is $J_{\beta,k}$, then the range of target number K is

$$J_{k-1} \leq K \leq J_{k-1} + J_{\beta,k} \times J_{k-1} + J_{\gamma,k} \tag{6.2}$$

As the measurements will be partitioned into several subsets, the number of subsets is set in a partition as the possible target number. To obtain different partitions, the different possible target number for measurement partitioning are used. For example, if there are J_{k-1} newborn targets and one target spawned from the same target, the partition number N_p is reduced to

$$N_p = 2 \times J_{k-1} + 1 \tag{6.3}$$

Each possible target, whether it's an existing one, a spawned one, or a newborn, has a matching subset in the partition.
Since the positions of spawned and newborn targets are unknown, their possible cluster centers are chosen using the K-means++ initialization method. This means measurements that are close together are grouped into the same subset.
To make sure the partition includes the most likely groupings, the predicted positions of

existing targets are used to find their possible cluster centers at the current time step. So, any measurement close to a predicted position is treated as a possible center. To make the partitioning more reliable, the algorithm first creates an initial partition using these predicted centers for the existing targets. Then it goes on with the rest of the process to refine the partition $P_1 = \{W^{(i)}\}_{i=1}^{J_{K-1}}$ is obtained by clustering, where $W^{(i)}$ denotes the i^{th} subset in the initial partition P_1. By comparing the expected number λ of measurements from the same target with the measured number $|W^{(i)}|$ of each subset, the unreasonable subsets and their corresponding target predictions can be found.
For each subset in P_1, if the condition $\frac{W(i)}{\lambda} \geq 2$ is checked, if this condition is satisfied, this means that it wrongly contains measurements of more than one target due to its clustering center. The corresponding target prediction of this clustering center is considered unreasonable.
Finally, using geometry, the positions of alternative points can be figured out based on how the predicted targets and measurements are spread out in space. By looking at this spatial distribution, the predicted state $x_{k|k-1}^{(i)}$ for the i^{th} unreasonable subset $W^{(i)}$ can be estimated as the geometric center of a polygon formed by the centers of actual measurement clusters.
To fix this, we replace the predicted target state $x_{k|k-1}^{(i)}$ with a more reasonable alternative point, one of the vertices of a regular polygon, based on the geometric layout. Then, the position $(p_{j,x},\, p_{j,y})$ of this alternative point can be calculated as follows

$$p_{j,x} = x_{k|k-1,x}^{(i)} + TH \times \cos\theta - \frac{2\pi(j-1)}{n_i} \tag{6.4}$$

$$p_{j,y} = x_{k|k-1,y}^{(i)} + TH \times \sin\theta - \frac{2\pi(j-1)}{n_i} \tag{6.5}$$

Where $\phi = \arctan\left(\frac{p_{1,y}-x_{k|k-1,y}^{(i)}}{p_{1,x}-x_{k|k-1,x}^{(i)}}\right)$ is the angle between the X-axis and the line passing through p_1 and $x_{k|k-1}^{(i)}$, TH is the distance between $x_{k|k-1}^{(i)}$ and the first alternative point which is decided according to the distribution of the distances between $x_{k|k-1}^{(i)}$ and the measurements, and $j = 2, ..., n_i$. As the number of polygon vertices, n_i is the expected target number of the underestimated subsets [58].

Bibliography

[1] R. Alonso, J. L. Crassidis, and J. L. Junkins. Vision-based relative navigation for formation flying of spacecraft. 2000. doi: 10.2514/6.2000-4439.

[2] Amarante, Sfair, and Winter. Stability and evolution of fallen particles around the surface of asteroid (101955) bennu. *Journal of Geophysical Research: Planets*, 496 (4):4319–4326, 2020.

[3] J. Aristoff, J. Horwood, N. Singh, A. B. Poore, C. Sheaff, and M. Jah. Multiple hypothesis tracking (mht) for space surveillance: Theoretical framework. *Advances in the Astronautical Sciences*, pages 55–74, 2014. doi: 10.2514/1.G003204.

[4] Y. P. Bach and M. Ishiguro. Thermal radiation pressure as a possible mechanism for losing small particles on asteroids. *Monthly Notices of the Royal Astronomical Society*, 512(3):4010–4019, 2021.

[5] Y. P. Bach and M. Ishiguro. Thermal radiation pressure as a possible mechanism for losing small particles on asteroids. *Monthly Notices of the Royal Astronomical Society*, 512(3):4010–4019, 2021.

[6] B. Bos, M. Ravine, M. Caplinger, and et al. Touch and go camera system (tagcams) for the osiris-rex asteroid sample return mission. *Space Science Reviews*, 214(1):37, 2018. doi: 10.1007/s11214-017-0465-2.

[7] G. M. Brown and D. J. Scheeres. Analyzing the structure of periodic orbit families that exist around asteroid (101955) bennu. *Celestial Mechanics and Dynamical Astronomy*, 134:41, 2022.

[8] J. A. Burns. Radiation forces on small particles in the solar system. *Icarus*, 40:1–48, 1979.

[9] S. Chesley and D. Farnocchia. Asteroid 101955 (1999 rq36) ephemeris delivery. *JPL Solution*, 76, 2013.

[10] S. R. Chesley, D. Farnocchia, M. C. Nolan, D. Vokrouhlický, P. W. Chodas, A. Milani, F. Spoto, B. Rozitis, L. A. M. Benner, W. F. Bottke, M. W. Busch, J. P. Emery,

E. S. Howell, D. S. Lauretta, J. L. Margot, and P. A. Taylor. Orbit and bulk density of the osiris-rex target asteroid (101955) bennu. *Icarus*, 235:5–22, 2014.

[11] S. R. Chesley, A. S. French, A. B. Davis, R. A. Jacobson, M. Brozović, D. Farnocchia, et al. Trajectory estimation for particles observed in the vicinity of (101955) bennu. *Journal of Geophysical Research: Planets*, 125(9):e2019JE006363, 2020.

[12] B. E. Clark, R. P. Binzel, E. S. Howell, E. A. Cloutis, A. M. Ockert-Bell, P. Christensen, and D. S. Lauretta. Asteroid (101955) 1999 rq36: Spectroscopy from 0.4 to 2.4 m and meteorite analogs. *Icarus*, 216(2):462–475, 2011.

[13] Colombo, Trisolini, and Tsuda. Ejecta dynamics around asteroids in view of in-orbit collection missions. *Journal of Guidance, Control, and Dynamics*, 45(12):2272–2289, 2022.

[14] L. Gao, M. Chen, and Q. Liu. Labeled gm-phd filter for explicit tracking of closely spaced targets. *Sensors*, 21(5):1654, 2021. doi: 10.3390/s21051654. URL `https://www.mdpi.com/1424-8220/21/5/1654`.

[15] Y. S. Garcia. Satellite orbit determination using sequential filtering methods. *M.Sc. Thesis*, 2022.

[16] A. Giannitrapani, F. Scortecci, N. Ceccarelli, and A. Garulli. Comparison of ekf and ukf for spacecraft localization via angle measurements. *IEEE Transactions on Aerospace and Electronic Systems*, 47(1):75–85, 2011. doi: 10.1109/TAES.2011.5705660.

[17] K. Granstrom, C. Lundquist, and U. Orguner. Extended target tracking using a gaussian-mixture phd filter. *IEEE Transactions on Aerospace and Electronic Systems*, 48(4):3268–3286, Oct 2012. doi: 10.1109/TAES.2012.6324703.

[18] W. D. Grossman. Novel line-of-sight three-dimensional triangulation algorithm for stationary targets. *Journal Of Guidance, Control, And Dynamics*, 2018. doi: 10.2514/1.G003204.

[19] V. E. Hamilton, A. A. Simon, P. R. Christensen, D. C. Reuter, B. E. Clark, M. A. Barucci, and D. S. Lauretta. Evidence for widespread hydrated minerals on asteroid (101955) bennu. *Nature Astronomy*, 3:332–340, 2019.

[20] K. Hata and S. Savarese. Camera models. *CS231A Course Notes 1*.

[21] C. W. Hergenrother, M. C. Nolan, P. P. Binzel, E. A. Cloutis, M. A. Barucci, P. Michel, and et al. Lightcurve, color and phase function photometry of the

osiris-rex target asteroid (101955) bennu. *Icarus*, 226(1):663–670, 2013. doi: 10.1016/j.icarus.2013.05.044.

[22] C. W. Hergenrother, M. A. Barucci, O. Barnouin, B. Bierhaus, R. P. Binzel, W. F. Bottke, S. Chesley, B. C. Clark, B. E. Clark, E. Cloutis, C. Drouet d'Aubigny, M. Delbo, J. Emery, B. Gaskell, E. Howell, L. Keller, M. Kelley, J. Marshall, P. Michel, M. Nolan, B. Rizk, D. Scheeres, D. Takir, D. Vokrouhlický, E. Beshore, and D. S. Lauretta. The design reference asteroid for the osiris-rex mission target (101955) bennu. *Icarus*, 226(1):663–670, 2014. doi: 10.1016/j.icarus.2013.12.015.

[23] C. W. Hergenrother, C. Maleszewski, J. Li, M. Pajola, S. R. Chesley, A. S. French, A. B. Davis, J. Y. Pelgrift, J. M. Leonard, F. Shelly, A. J. Liounis, K. Becker, S. S. Balram-Knutson, R. Garcia, T. R. Kareta, C. Adam, K. Alkiek, B. J. Bos, M. Brozović, K. N. Burke, E. Christensen, B. E. Clark, D. N. DellaGiustina, C. D. d'Aubigny, D. Farnocchia, E. S. Howell, R. A. Jacobson, J. N. Kidd, E. J. Lessac-Chenen, R. Melikyan, M. C. Nolan, R. S. Park, S. Selznick, B. Rizk, and D. S. Lauretta. Photometry of particles ejected from active asteroid (101955) bennu. *Journal of Geophysical Research: Planets*, 125(3):e2019JE006349, 2020.

[24] S. J. Iulier and J. K. Uhlmann. A new extension of the kalman filter to non linear filters. 1996.

[25] R. E. KALMAN. A new approach to linear filtering and prediction problems. *Transactions of the ASME*, 1960.

[26] W. M. Kaula. *Theory of Satellite Geodesy.* Blaisdell Publishing Company, 1966.

[27] M. Küppers, L. O'Rourke, D. Bockelée-Morvan, and et al. Localized sources of water vapour on the dwarf planet (1) ceres. *Nature*, 505(7484):525–527, 2014. doi: 10.1038/nature12918. URL `https://doi.org/10.1038/nature12918`.

[28] K. Langner, F. Marzari, A. Rossi, and G. Zanotti. Long-term dynamics around the didymos–dimorphos binary asteroid of boulders ejected after the dart impact. *Astronomy & Astrophysics*, 670:A153, 2023.

[29] K. Langner, F. Marzari, A. Rossi, and G. Zanotti. Long-term dynamics around the didymos–dimorphos binary asteroid of boulders ejected after the dart impact. *Astronomy & Astrophysics*, 670:A153, 2023.

[30] D. Lanza. Asteroid identification in deep space: Autonomous detection and association with computer vision, 2024. Laurea Magistrale in Space Engineering - Ingegneria Spaziale, Politecnico di Milano, Academic Year 2023–2024.

[31] J. N. Larson and G. Sarid. An n-body approach to modelling debris and ejecta off small bodies: implementation and application. *Monthly Notices of the Royal Astronomical Society*, 506(4):5021–5033, 2021.

[32] D. S. Lauretta, D. N. Dellagiustina, C. A. Bennett, D. R. Golish, K. J. Becker, S. S. Balram-Knutson, et al. The unexpected surface of asteroid (101955) bennu. *Nature*, 568:55–60, 2019.

[33] D. S. Lauretta, C. W. Hergenrother, S. R. Chesley, J. M. Leonard, J. Y. Pelgrift, C. D. Adam, and et al. Episodes of particle ejection from the surface of the active asteroid (101955) bennu. *Science*, 366(6470):eaay3544, 2019. doi: 10.1126/science.aay3544.

[34] J. M. Leonard, C. D. Adam, J. Y. Pelgrift, E. J. Lessac-Chenen, D. S. Nelson, P. G. Antreasian, A. J. Liounis, M. C. Moreau, C. W. Hergenrother, S. R. Chesley, M. C. Nolan, and D. S. Lauretta. Initial orbit determination and event reconstruction from estimation of particle trajectories about (101955) bennu. *Earth and Space Science*, 7, 2020. doi: 10.1029/2020JE006381.

[35] Liounis. Giant documentation.

[36] A. J. Liounis, J. L. Small, J. C. Swenson, J. R. Lyzhoft, B. W. Ashman, K. M. Getzandanner, M. C. Moreau, J. M. Leonard, S. R. Chesley, D. S. Nelson, C. W. Hergenrother, C. D. Adam, J. Y. Pelgrift, and B. J. Barbee. Autonomous detection of particles and tracks in optical images. *Earth and SPace Science*, 2020. doi: https://doi.org/10.1029/2019ea000843.

[37] J.-C. Liu, H. A. Zook, and A. A. Jackson. Radiation pressure, poynting-robertson drag, and solar wind drag in the restricted three body problem. *Icarus*, 116:186–201, 1995.

[38] R. J. Macke, G. J. Consolmagno, and D. T. Britt. Density, porosity, and magnetic susceptibility of carbonaceous chondrites. *Meteoritics & Planetary Science*, 46(12): 1842–1862, 2011.

[39] P. C. Mahalanobis. On the generalized distance in statistics. *Proceedings of the National Institute of Sciences (Calcutta)*, 2:49–55, 1936.

[40] H. Masnadi-Shirazi, A. Masnadi-Shirazi, and M.-A. Dastgheib. A step by step mathematical derivation and tutorial on kalman filters. *Transactions of the ASME*, 2019.

[41] E. Mazarico, D. E. Smith, and R. A. Taylor. Dynamics of simulated ejecta particles around small solar system bodies. Technical Report NASA/TM-2020-123456, NASA

Technical Reports Server, 2020. URL `https://ntrs.nasa.gov/api/citations/202000123456.pdf`.

[42] J. W. McMahon, D. J. Scheeres, S. G. Hesar, D. Farnocchia, S. Chesley, and D. Lauretta. The osiris-rex radio science experiment at bennu. *Space Science Reviews*, 214 (3):53, 2018. doi: 10.1007/s11214-018-0492-5.

[43] J. W. McMahon, D. J. Scheeres, S. R. Chesley, A. French, D. Brack, D. Farnocchia, et al. Dynamical evolution of simulated particles ejected from asteroid bennu. *Journal of Geophysical Research: Planets*, 125:e2019JE006229, 2020. doi: 10.1029/2019JE006229.

[44] F. Mignard. Effects of radiation forces on dust particles in planetary rings. *IAU Colloq. 75: Planetary Rings*, pages 333–366, 1984.

[45] M. C. Nolan, C. Magri, E. S. Howell, L. A. M. Benner, J. D. Giorgini, C. W. Hergenrother, R. S. Hudson, D. S. Lauretta, J. L. Margot, and S. J. Ostro. Shape model and surface properties of the osiris-rex target asteroid (101955) bennu from radar and lightcurve observations. *Icarus*, 226:629–640, 2013.

[46] R. Parihar, S. Sarkar, S. Vora, and R. V. Babu. Monoplace3d: Learning 3d-aware object placement for 3d monocular detection. 2025. doi: 10.48550/arXiv.2504.06801.

[47] J. Y. Pelgrift, E. J. Lessac-Chenen, C. D. Adam, J. M. Leonard, D. S. Nelson, L. McCarthy, E. M. Sahr, A. Liounis, M. Moreau, and B. J. Bos. Reconstruction of bennu particle events from sparse data. *Earth and Space Science*, 7, 2020. doi: 10.1029/2019EA000938. URL `https://doi.org/10.1029/2019EA000938`.

[48] B. Rizk, C. Drouet d'Aubigny, D. Golish, and et al. Ocams: The osiris-rex camera suite. *Space Science Reviews*, 214(1):26, 2018. doi: 10.1007/s11214-017-0460-7.

[49] A. Rizza, C. Buonagura, P. Panicucci, and F. Topputo. Modular pipeline for small bodies gravity field modeling: an efficient representation of variable density spherical harmonics coefficients. *Advances in Space Research*, 2024.

[50] Scheeres and Ostro. Orbits close to asteroid 4769 castalia. *Icarus*, 121(1):67–87, 1995.

[51] D. Scheeres. Orbital mechanics about small bodies. *Acta Astronautica*, 72:1–14, 2012.

[52] D. Scheeres. Orbital mechanics about small asteroids. *Acta Astronautica*, 102:290–295, 2014.

[53] D. Scheeres. Orbital mechanics about asteroids and comets. *Journal of Guidance, Control, and Dynamics*, 39(1):22–38, 2016.

[54] D. J. Scheeres, J. W. McMahon, A. S. French, D. N. Brack, S. R. Chesley, D. Farnocchia, and D. S. Lauretta. The dynamic geophysical environment of (101955) bennu based on osiris-rex measurements. *Nature Astronomy*, 3:352–361, 2019.

[55] D. J. Scheeres, A. S. French, and P. Tricarico. Heterogeneous mass distribution of the rubble-pile asteroid (101955) bennu. *Science Advances*, 6(41):eabc3350, 2020.

[56] S. Segal, P. Gurfil, and A. Carmi. Stereovision-based estimation of relative dynamics between noncooperative satellites: Theory and experiments. *Transactions on Control Systems Technology*, 2014. doi: 10.1109/TCST.2013.2255288.

[57] N. Singh, J. T. Horwood, J. rey M. Aristo, A. B. Poore, C. Sheaff, and M. K. Jah. Multi hypothesis tracking (mht) for space surveillance results and simulation studies. 2013.

[58] L. Sun, H. Yu, Z. Fu, Z. He, and F. Tao. Tracking of multiple closely spaced extended targets based on prediction-driven measurement sub-partitioning algorithm. *Applied Sciences*, 10(14):4936, 2020. doi: 10.3390/app10144936. URL `https://www.mdpi.com/2076-3417/10/14/4936`.

[59] F. Tao, L. Sun, and Z. He. Multi-feature matching gm-phd filter for robust multi-target tracking. *IEEE Access*, 10:713–726, 2022. doi: 10.1109/ACCESS.2022.3141592. URL `https://ieeexplore.ieee.org/document/3141592`.

[60] Topputo and D. Lizia. Spacecraft guidance and navigation course.

[61] D. A. Vallado and W. D. McClain. *Fundamentals of Astrodynamics and Applications*. Microcosm Press, 4th edition, 2013. ISBN 9781881883180. URL `https://microcosmpress.com/publishing/fundamentals-of-astrodynamics-and-applications-fourth-edition/`.

[62] B.-N. Vo and W.-K. Ma. The gaussian mixture probability hypothesis density filter. *IEEE TRANSACTIONS ONSIGNALPROCESSING*, 54(11), 2006.

[63] X. Wang, Y. Li, and Z. Zhang. Refined phd filter under low detection probability. *IEEE Transactions on Aerospace and Electronic Systems*, 55(3):1201–1212, 2019. doi: 10.1109/TAES.2019.1234567. URL `https://ieeexplore.ieee.org/document/1234567`.

[64] Z. Wang, J. Yi, A. Chen, and G. Han. Movis: When 3d object detection is like

human monocular vision. *IEEE Transactions on Image Processing*, 2025. doi: 10.1109/TIP.2025.3544880.

[65] W. Yin, Y. Shi, L. Shu, and Y. Gao. Autonomous navigation of an asteroid orbiter enhanced by a beacon satellite in a high-altitude orbit. *Astrodynamics*, 8(3):375–400, 2024. doi: 10.1007/s42064-023-0172-6.

A | Derivation of the Equations of Motion

The appendix is an in depth review of how the equations of motion around Bennu have been derived. In the end the used integrator specifics are shown. The used harmonic coefficients can also be found.

A.1. Spherical Harmonic Representation of the Gravity Field Potential

Appendix A.1. contains the mathematical derivation for expressing the gravity field as an harmonic expansion. An algorithm for computing recursively the coefficient for a faster computer implementation will be also described. The used approach follows the steps described in [61].

A.1.1. Gravitational Potential

The gravitational potential of a body can be modeled as an expansion in spherical harmonics as

$$U = \frac{\mu}{r}\left\{1 + \sum_{n=1}^{\infty}\sum_{m=0}^{n}\left(\frac{R_{\text{ref}}}{r}\right)^n P_{nm}(\sin\varphi)\left[C_{nm}\cos(m\lambda) + S_{nm}\sin(m\lambda)\right]\right\} \tag{A.1}$$

where μ is the product of the universal constant of gravitation G and the mass of the main body M, a_e is the semi-major axis of the body's reference ellipsoid, and r, φ, λ are the particle's distance, latitude, and longitude, respectively, in a body-fixed coordinate system. C_{nm} and S_{nm} are spherical harmonic coefficients of degree n and order m, and P_{nm} are the associated Legendre functions of degree n and order m. A gravitational model consists of a set of constants that specify a_e and the C_{nm}, S_{nm} coefficients. The geopotential can be thought of as consisting of three constituent parts

$$U = U_0 + U_1 + U_2 \tag{A.2}$$

The first coefficient is the expansion term corresponding to the degree and order zero term. The Associated Legendre Function, P_{00} and the C_{00} coefficient are then equal to one, bringing to

$$U_0 = \frac{\mu}{r} \tag{A.3}$$

The second coefficient of the spherical harmonic representation are the terms corresponding to $m = 0$, representing the zonal contribution to the potential

$$U_1 = \frac{\mu}{r} \sum_{\ell=1}^{\infty} (\frac{a_e}{r})^l P_{l,0}(\sin\phi) C_{l,0} \tag{A.4}$$

The remaining part of the spherical harmonic representation is that part depending on longitude

$$U_2 = \frac{\mu}{r} \sum_{\ell=1}^{\infty} \left(\frac{a_e}{r}\right)^l \sum_{m=1}^{\infty} P_{l,m}(\sin\phi) \left[C_{l,m} \cos m\lambda + C_{l,m} \sin m\lambda\right] \tag{A.5}$$

Thus, eq. A.1 can be rewritten as

$$\begin{aligned} U = \frac{\mu}{r} + \frac{\mu}{r} \sum_{\ell=1}^{\infty} \left(\frac{a_e}{r}\right)^\ell P_{\ell,0}(\sin\phi) C_{\ell,0} \\ + \frac{\mu}{r} \sum_{\ell=1}^{\infty} \left(\frac{a_e}{r}\right)^\ell \sum_{m=1}^{\infty} P_{\ell,m}(\sin\phi) \left[C_{\ell,m} \cos(m\lambda) + S_{\ell,m} \sin(m\lambda)\right] \end{aligned} \tag{A.6}$$

A.1.2. Spherical Harmonics

The spherical harmonics function are formed by the product of the Associated Legendre Functions with $\cos m\lambda$ and $\sin m\lambda$

$$A_{\ell,m}(\phi,\lambda) = P_{\ell,m}(\sin\phi)\cos m\lambda \quad \text{and} \quad B_{\ell,m}(\phi,\lambda) = P_{\ell,m}(\sin\phi)\sin m\lambda \tag{A.7}$$

These coefficients need to be normalized as follows

$$\bar{P}_{l,m} = \left[(2-\delta_{m0})(2l+1)\frac{(l-m)!}{(l+m)!}\right]^{1/2} P_{l,m} \tag{A.8}$$

where

$$\delta_{mn} = \begin{cases} 1 & \text{if } m = 0 \\ 0 & \text{if } m > 0 \end{cases} \tag{A.9}$$

The geopotential coefficients $C_{l,m}$ and $S_{l,m}$ are normalized by the inverse of this scale factor

$$\left\{ \begin{array}{c} \bar{C}_{l,m} \\ \bar{S}_{l,m} \end{array} \right\} = \left[\frac{1}{(2-\delta_{m0})(2l+1)}\frac{(l+m)!}{(l-m)!}\right]^{1/2} \left\{ \begin{array}{c} C_{l,m} \\ S_{l,m} \end{array} \right\} \tag{A.10}$$

Eq. A.1 can now be written in normalized coordinates

$$U = \frac{\mu}{r}\left\{1 + \sum_{n=1}^{\infty}\sum_{m=0}^{n}\left(\frac{R_{\text{ref}}}{r}\right)^{n} \overline{P}_{nm}(\sin\varphi)\left[\overline{C}_{nm}\cos(m\lambda) + \overline{S}_{nm}\sin(m\lambda)\right]\right\} \tag{A.11}$$

A.1.3. Associated Legendre Functions

In general the Associated Legendre Function of degree and order m is

$$P_{l,m}(x) = \frac{(1-x^2)^{m/2}}{2^l}\sum_{k=0}^{\frac{l-m}{2}}(-1)^k\frac{(2l-2k)!}{k!(l-k)!(l-m-2k)!}x^{l-m-2k} \tag{A.12}$$

Usually these functions are computed recursively, the starting values for the recursion are the $l = m$ and $l = m + 1$ functions, computed from

$$P_{l,l}(x) = \frac{(2l-1)!}{2^{l-1}(l-1)!}(1-x^2)^{l/2} \tag{A.13}$$

$$P_{l,l-1}(x) = \frac{x}{(1-x^2)^{1/2}}P_{l,l}(x) \tag{A.14}$$

Then one possible recursion is to compute $P_{l,l}$ and $P_{l,l-1}$, and then compute the functions for all lower orders of degree using

$$P_{l,m}(x) = \frac{1}{(l-m)(l+m+1)}\left[2(m+1)\frac{x}{(1-x^2)^{1/2}}P_{l,m+1}(x) - P_{l,m+2}(x)\right] \tag{A.15}$$

A.1.4. Gravitational Acceleration

The gravitational acceleration at a specific location is determined by calculating the gradient of the potential. The use of a body-fixed frame is advantageous for determining the gravity field of the body because the gravity coefficients are constant and their basis functions are orthogonal in such a frame. In the body-centered body-fixed spherical coordinate system, the gradient is expressed as

$$\vec{a} = a_r\vec{u}_r + a_\phi\vec{u}_\phi + a_\lambda\vec{u}_\lambda \tag{A.16}$$

$$\vec{a} = \nabla U = \frac{\partial U}{\partial r}\vec{u}_r + \frac{1}{r}\frac{\partial U}{\partial \phi}\vec{u}_\phi + \frac{1}{r\cos\phi}\frac{\partial U}{\partial \lambda}\vec{u}_\lambda \tag{A.17}$$

where $\vec{u}_r$, $\vec{u}_\phi$, and $\vec{u}_\lambda$ are unit vectors in the r, ϕ, λ reference frame. This basis has $\vec{u}_r$ pointing along the radius vector to the orbiting body, $\vec{u}_\phi$ in the direction of increasing north latitude, and $\vec{u}_\lambda$ in the direction of increasing east longitude. Following, the components of the inertial acceleration in the body-fixed (rotating) coordinate system are obtained. Substituting for the gravitational potential (Eq. A.1) and taking the indicated partials in (Eq. A.16) gives the acceleration vector

$$\begin{aligned}
\vec{a} = &\left\{-\frac{\mu}{r^2}\sum_{\ell=0}^{\infty}(\ell+1)\left(\frac{a_e}{r}\right)^\ell \sum_{m=0}^{\ell} P_{\ell,m}(\sin\phi)[C_{\ell,m}\cos m\lambda + S_{\ell,m}\sin m\lambda]\right\}\vec{u}_r \\
&+\left\{\frac{\mu}{r^2}\sum_{\ell=1}^{\infty}\left(\frac{a_e}{r}\right)^\ell \sum_{m=0}^{\ell} \frac{\partial P_{\ell,m}(\sin\phi)}{\partial\phi}[C_{\ell,m}\cos m\lambda + S_{\ell,m}\sin m\lambda]\right\}\vec{u}_\phi \\
&+\left\{\frac{\mu}{r^2}\sum_{\ell=1}^{\infty}\left(\frac{a_e}{r}\right)^\ell \sum_{m=1}^{\ell} m\frac{P_{\ell,m}(\sin\phi)}{\cos\phi}[-C_{\ell,m}\sin m\lambda + S_{\ell,m}\cos m\lambda]\right\}\vec{u}_\lambda
\end{aligned} \tag{A.18}$$

The leading term of the radial component (degree and order equal to zero) is simply the expected two-body gravitational acceleration $-\mu/r^2$. Next, the body-fixed Cartesian components of the acceleration can be obtained by rotating from the spherical coordinates to the x, y, z basis.

A.2. Equations of Motion

When all the contributions to the acceleration have been computed the orbit propagation around the asteroid can be computed with the following equation of motion

$$\begin{bmatrix} \frac{d\mathbf{r}}{dt} \\ \frac{d\mathbf{v}}{dt} \end{bmatrix} = \begin{bmatrix} v \\ a_{2-body} + \mathbf{T}_{AI} a_{harmonic} + a_{SRP} \end{bmatrix} \tag{A.19}$$

where r and v denote the position and velocity vectors, respectively. The term a_{2-body} refers to the two-body acceleration, which is expressed as

$$a_{2-body} = -\frac{\mu}{r^3}\mathbf{r} \tag{A.20}$$

The terms a_{srp} and $a_{harmonic}$ are the accelerations owing to the SRP and the non spherical gravity perturbation respectively. Moreover, $\mathbf{T}_{AI}$ is the transformation matrix from spherical to inertial coordinates

$$\mathbf{T}_{AI} = \mathbf{R3}(-(90+\alpha_0))\mathbf{R1}(-(90-\delta_0))\mathbf{R3}(-W) \tag{A.21}$$

where **R1** and **R3** are the standard rotation matrices for right-handed rotations around the x- and z- axis respectively, α_0 and δ_0 are defined in **??** While the angle W between the $x-axis$ of asteroid-centered asteroid equator inertial reference frame and crossing line of the asteroid and Earths equators at the J2000 epoch (JD 2451545.0) [65].

A.3. Numerical Integration

In order to simulate the motion of the particles and the spacecraft around the gravitational body of interest, the before mentioned equation of motions are integrated with the seventh order Runge-Kutta method $ODE78$ in MATLAB. The relative tolerance was set to $1e^{-12}$, while the absolute tolerance was kept to $1e^{-8}$ and $1e^{-11}$ for the position and the velocity respectively. The propagation is carried out in the time span considered in the simulation, or when the particles impacts the main orbiting body.

A.4. System Time Discretization

The equation of motion are integrated as explained before, because of the high number of operations that are required on the obtained trajectories each one is discretized at the measurements time instants.

In particular, the ODE function produces the variable "object", containing the propagation information. Then the discretizing time span is built considering the start time of the simulation, the measurement cadence, and the end time of the simulation, or the actual stopping time, in case an event function stops the propagation before the actual end of the simulation.

Table A.1: Estimated Spherical Harmonic Coefficients (Normalized) With Uncertainty

Coefficient	Estimated Value
J_2	$1.926 \times 10^{-2} \pm 5.2 \times 10^{-5}$
$C_{2,2}$	$3.06 \times 10^{-3} \pm 5.3 \times 10^{-5}$
$S_{2,2}$	$-1.09 \times 10^{-4} \pm 1.1 \times 10^{-4}$
J_3	$-1.22 \times 10^{-3} \pm 7.9 \times 10^{-5}$
$C_{3,1}$	$8.15 \times 10^{-4} \pm 2.9 \times 10^{-5}$
$S_{3,1}$	$-5.43 \times 10^{-4} \pm 4.3 \times 10^{-5}$
$C_{3,2}$	$-9.33 \times 10^{-4} \pm 4.6 \times 10^{-5}$
$S_{3,2}$	$-5.38 \times 10^{-4} \pm 4.9 \times 10^{-5}$
$C_{3,3}$	$1.17 \times 10^{-3} \pm 6.0 \times 10^{-5}$
$S_{3,3}$	$-3.1 \times 10^{-4} \pm 8.3 \times 10^{-5}$
J_4	$-6.50 \times 10^{-4} \pm 7.1 \times 10^{-5}$
$C_{4,1}$	$-8.82 \times 10^{-5} \pm 5.8 \times 10^{-5}$
$S_{4,1}$	$-5.8 \times 10^{-4} \pm 6.0 \times 10^{-5}$
$C_{4,2}$	$-8.71 \times 10^{-4} \pm 5.5 \times 10^{-5}$
$S_{4,2}$	$-8.4 \times 10^{-5} \pm 6.5 \times 10^{-5}$
$C_{4,3}$	$-7.6 \times 10^{-4} \pm 6.8 \times 10^{-5}$
$S_{4,3}$	$-3.9 \times 10^{-4} \pm 6.3 \times 10^{-5}$
$C_{4,4}$	$7.7 \times 10^{-5} \pm 1.6 \times 10^{-4}$
$S_{4,4}$	$2.25 \times 10^{-3} \pm 8.5 \times 10^{-5}$
J_5	$6.7 \times 10^{-5} \pm 1.0 \times 10^{-4}$
$C_{5,1}$	$-3.5 \times 10^{-4} \pm 8.4 \times 10^{-5}$
$S_{5,1}$	$1.6 \times 10^{-4} \pm 7.9 \times 10^{-5}$
$C_{5,2}$	$-3.7 \times 10^{-4} \pm 8.1 \times 10^{-5}$
$S_{5,2}$	$-2.7 \times 10^{-4} \pm 7.4 \times 10^{-5}$
$C_{5,3}$	$-2.2 \times 10^{-4} \pm 8.2 \times 10^{-5}$
$S_{5,3}$	$-9.5 \times 10^{-4} \pm 7.5 \times 10^{-5}$
$C_{5,4}$	$3.2 \times 10^{-4} \pm 7.2 \times 10^{-5}$
$S_{5,4}$	$5.0 \times 10^{-4} \pm 8.2 \times 10^{-5}$
$C_{5,5}$	$-2.3 \times 10^{-4} \pm 8.7 \times 10^{-5}$
$S_{5,5}$	$3.0 \times 10^{-4} \pm 7.3 \times 10^{-5}$
J_6	$1.37 \times 10^{-3} \pm 8.2 \times 10^{-5}$
J_7	$-7.9 \times 10^{-5} \pm 1.2 \times 10^{-4}$
J_8	$-6.8 \times 10^{-4} \pm 1.1 \times 10^{-4}$
J_9	$-3.5 \times 10^{-5} \pm 1.0 \times 10^{-4}$
J_{10}	$2.0 \times 10^{-4} \pm 9.9 \times 10^{-5}$

Note: Coefficients computed using the orbits of well known particles. The reference radius is 290 m. All coefficients are estimated through degree 10, but sectorial and tesseral coefficients are tabulate only through degree 5 and zonal coefficients through degree 10 [11]. In [42] another set of parameters, computed before the arrival of OSIRIS-Rex, can be found.

B | Linear Motion Assumption Radiant Point and States Derivation

The appendix contains the equations used to compute the radiant point and the particle states following the approach of [47], following a linear motion assumption of the particle motion, as already described in 3.2.

B.1. Derivation

B.1.1. Nomenclature

- **Radiant:** The ejection location (near or far solution).
- **ImagePlane:** Plane perpendicular to the camera's boresight that contains the ejection location.
- **Anchor:** Closest point to the spacecraft in the image plane along the particle's observed line.
- t_i: Time of the i-th observation.
- l_i: Observation location in the image (relative to the radiant).
- α_i: Angular displacement from anchor at time t_i.
- a: Distance from anchor to radiant.
- R: Distance from spacecraft to anchor.
- $\vec{r}_i$: 3D vector from spacecraft to particle at t_i.
- $\vec{r}_{sc-l_i}$: 3D vector from spacecraft to observation point l_i.
- x_i: In-plane component of particle position relative to radiant.

- y_i: Out-of-plane component of particle position.
- ω_{ij}: Observation angle rate between times t_i and t_j.
- t_0: Ejection time (epoch).

Subscripts

- i = i th detection epoch index
- 0 = radiant epoch index

Visually these quantities can be found in plot B.1.

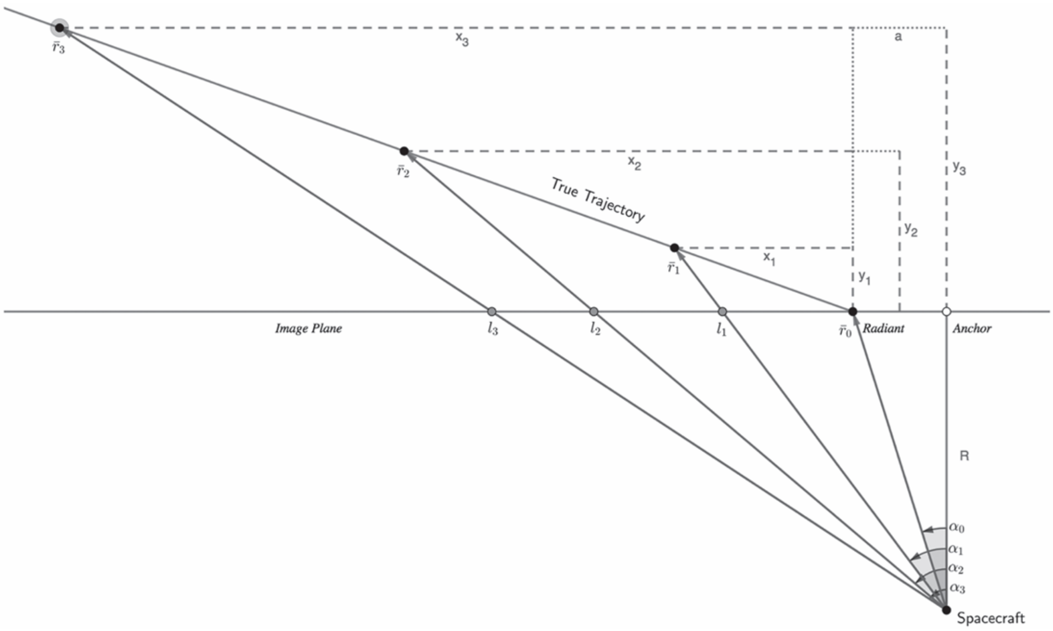


Figure B.1: The linear geometry that results from assuming that the particles' velocities remain constant. A particle following a linear trajectory connecting the 3-D positions. $r_{1,2,3}$ appears with in the image at the locations $l_{1,2,3}$ when observed at the times $t_{1,2,3}$. The x and y components of each $3-D$ position with respect to the radiant point are also shown where the x direction is parallel to the line connecting the observations with in the image plane, and they direction is parallel to the camera bore sight. The z direction completes the right-hand rule,such that the z component of each position is zero. Plot taken from [42]

.

B.1.2. Equations

(1) Similar triangles:

$$\frac{x_i + a}{y_i + R} = \frac{l_i + a}{R} \tag{B.1}$$

(2–3) Constant 3D velocity:

$$\frac{y_j}{t_j - t_0} = \frac{y_i}{t_i - t_0}, \qquad \frac{x_j}{t_j - t_0} = \frac{x_i}{t_i - t_0} \tag{B.2}$$

Number of unknowns: $2n + 1$
Number of knowns: $n + 2(n - 1) = 3n - 2$

- $n = 2$: underconstrained
- $n = 3$: deterministic
- $n > 3$: overconstrained

B.1.3. Ejection Epoch, 2 Observations

Assuming constant observation angle rate:

$$\frac{l_j - l_i}{t_j - t_i} = \frac{l_i}{t_i - t_0} \tag{B.3}$$

Solving for t_0:

$$t_0 = t_i - \frac{l_i}{l_j - l_i}(t_j - t_i) \tag{B.4}$$

B.1.4. Ejection Epoch Given Three or More Observations

From equation (1),

$$R(x_i + a) = (l_i + a)(y_i + R) \tag{B.5}$$

$$Rx_i + Ra = l_i y_i + a y_i + l_i R + aR \tag{B.6}$$

$$x_i = \frac{(l_i + a)}{R} y_i + l_i \tag{B.7}$$

Now, letting $\tau_i = t_i - t_0$, equations (2) and (3) become:

$$\frac{y_i}{y_j} = \frac{\tau_i}{\tau_j} \tag{B.8}$$

$$\frac{x_i}{x_j} = \frac{\tau_i}{\tau_j} \tag{B.9}$$

Substituting equation (6) into equation (8):

$$\frac{\frac{(l_i+a)}{R}y_i + l_i}{\frac{(l_j+a)}{R}y_j + l_j} = \frac{\tau_i}{\tau_j} \tag{B.10}$$

$$\frac{(l_i + a)y_i + l_iR}{(l_j + a)y_j + l_jR} = \frac{\tau_i}{\tau_j} \tag{B.11}$$

$$\frac{m_iy_i + n_i}{m_jy_j + n_j} = \frac{\tau_i}{\tau_j} \tag{B.12}$$

Setting $j = 1$, defining:

$$m_i = l_i + 1, \quad n_i = l_iR, \quad \delta_i = \frac{\tau_i}{\tau_1}$$

Then:

$$\frac{m_iy_i + n_i}{m_jy_j + n_j} = \frac{\tau_i}{\tau_j} \tag{B.13}$$

Setting $j = 1$, and letting $\delta_i = \frac{\tau_i}{\tau_1}$, the equation becomes:

$$\frac{m_iy_i + n_i}{m_1y_1 + n_1} = \delta_i \tag{B.14}$$

From equation (6), we also have:

$$y_i = \delta_iy_1$$

Combining (10) and (11):

$$\frac{m_i\delta_iy_1 + n_i}{m_1y_1 + n_1} = \delta_i \tag{B.15}$$

$$m_i\delta_iy_1 + n_i = m_1\delta_iy_1 + n_1\delta_i \tag{B.16}$$

$$(m_i - m_1)\delta_iy_1 = n_1\delta_i - n_i \tag{B.17}$$

$$y_1 = \frac{n_1\delta_i - n_i}{(m_i - m_1)\delta_i} \tag{B.18}$$

Let $i = 2, 3$ and equate

$$\frac{n_1\delta_2 - n_2}{(m_2 - m_1)\,\delta_2} = \frac{n_1\delta_3 - n_2}{(m_3 - m_1)\,\delta_3} \tag{B.19}$$

$$(n_1\delta_2 - n_2)(m_3 - m_1)\,\delta_3 = (n_1\delta_3 - n_3)(m_2 - m_1)\,\delta_2 \tag{B.20}$$

$$n_1(m_3 - m_1)\,\delta_2\delta_3 \;-\; n_2(m_3 - m_1)\,\delta_3 \;=\; n_1(m_2 - m_1)\,\delta_2\delta_3 \;-\; n_3(m_2 - m_1)\,\delta_2 \tag{B.21}$$

$$n_1(m_3 - m_2)\,\delta_2\delta_3 \;+\; n_3(m_2 - m_1)\,\delta_2 \;-\; n_2(m_3 - m_1)\,\delta_3 \;=\; 0 \tag{B.22}$$

$$n_1(m_3 - m_2)\frac{\tau_2\tau_3}{\tau_1^2} + n_3(m_2 - m_1)\frac{\tau_2}{\tau_1} - n_2(m_3 - m_1)\frac{\tau_3}{\tau_1} = 0 \tag{B.23}$$

$$n_1(m_3 - m_2)\tau_2\tau_3 + n_3(m_2 - m_1)\tau_2\tau_1 - n_2(m_3 - m_1)\tau_3\tau_1 = 0 \tag{B.24}$$

Expanding the terms

$$\tau_i\tau_J = (t_i - t_0)(t_j - t_0) \tag{B.25}$$

$$= t_i t_j - (t_i + t_j)t_0 + t_0^2 \tag{B.26}$$

Substitute into the previous relation and introduce

$$k_{ij} = m_i - m_j, \quad T_{ij} = t_i\, t_j, \quad q_{ij} = t_i + t_j,$$

to obtain

$$n_1k_{32}\,(T_{32} - q_{32}t_0 + t_0^2) + n_3k_{21}\,(T_{21} - q_{21}t_0 + t_0^2) + n_2k_{31}\,(T_{31} - q_{31}t_0 + t_0^2) = 0.$$

Collecting powers of t_0 yields a quadratic

$$a\,t_0^2 \;+\; b\,t_0 \;+\; c \;=\; 0, \tag{B.27}$$

with coefficients

$$a = n_1k_{32} + n_3k_{21} - n_2k_{31} = R\big[l_1l_3 - l_1l_2 + l_3l_2 - l_3l_1 - l_2l_3 + l_2l_1\big] = 0,$$
$$b = n_2k_{31}q_{31} - n_1k_{32}q_{32} - n_3k_{21}q_{21} = R\big[l_1(l_3 - l_2)\,t_1 + l_2(l_1 - l_3)\,t_2 + l_3(l_2 - l_1)\,t_3\big],$$
$$c = n_1k_{32}T_{32} + n_3k_{21}T_{21} - n_2k_{31}T_{31} = R\big[l_1(l_3 - l_2)\,t_3t_2 + l_2(l_1 - l_3)\,t_3t_1 + l_3(l_2 - l_1)\,t_2t_1\big].$$

(B.28)

(B.29)

(B.30)

Since $a = 0$, the linear solution is

$$b\,t_0 + c = 0 \quad \Longrightarrow \quad t_0 = -\frac{c}{b} = -\frac{R\big[l_1(l_3 - l_2)\,t_3t_2 + l_2(l_1 - l_3)\,t_3t_1 + l_3(l_2 - l_1)\,t_2t_1\big]}{R\big[l_1(l_3 - l_2)\,t_1 + l_2(l_1 - l_3)\,t_2 + l_3(l_2 - l_1)\,t_3\big]}$$

and simplifying the common R gives

$$\boxed{t_0 = -\,\frac{l_1(l_3 - l_2)\,t_3t_2 \; + \; l_2(l_1 - l_3)\,t_3t_1 \; + \; l_3(l_2 - l_1)\,t_2t_1}{l_1(l_3 - l_2)\,t_1 \; - \; l_2(l_1 - l_3)\,t_2 \; + \; l_3(l_2 - l_1)\,t_3}}\,.$$

This can be put into a convenient matrix form:

$$A = \begin{bmatrix} t_3t_2 & t_3t_1 & t_2t_1 \end{bmatrix} \quad B = \begin{bmatrix} t_1 & t_2 & t_3 \end{bmatrix} \quad C = \begin{bmatrix} l_1(l_3 - l_2) \\ l_2(l_1 - l_3) \\ t_3(l_2 - l_1) \end{bmatrix} \tag{B.31}$$

$$t_0 = \frac{-AB}{BC} \tag{B.32}$$

B.1.5. Three-Dimensional Positions Given Ejection Epoch and Three Observations

Taking equation (4) for epochs i and j and dividing and substituting equation (6),

$$\frac{x_j}{x_i} = \frac{(l_j + a)y_j + l_jR}{(l_i + a)y_i + l_iR} \tag{B.33}$$

$$\frac{\tau_j}{\tau_i} = \frac{(l_j + a)y_j + l_jR}{(l_i + a)y_i + l_iR} \tag{B.34}$$

$$(l_i + a)\tau_j y_i + l_i\tau_j R = (l_j + a)\tau_i y_j + l_j\tau_i R \tag{B.35}$$

$$(l_i + a)\tau_j y_i + (l_j + a)\tau_i y_j = (l_j\tau_i - l_i\tau_j)R \tag{B.36}$$

$$(l_i + a)\tau_j y_i + (l_j + a)\tau_i \cdot \frac{\tau_j}{\tau_i} y_j = (l_j\tau_i - l_i\tau_j)R \tag{B.37}$$

$$(l_i - l_j)\tau_j y_i = (l_j\tau_i - l_i\tau_j)R \tag{B.38}$$

$$y_i = -\left[\frac{\left(\frac{\tau_i}{\tau_j} l_j - l_i\right)}{l_j - l_i}\right] R \tag{21}$$

Constructing the spacecraft-to-particle position vector:

$$\bar{r}_i = (R + y_i)\frac{1}{\cos\alpha_i}\hat{r}_i \tag{B.39}$$

$$= \frac{l_j\left(1 - \frac{\tau_i}{\tau_j}\right)}{l_j - l_i} R \cdot \frac{1}{\cos\alpha_i}\hat{r}_i \tag{B.40}$$

$$= \frac{l_j\left(1 - \frac{\tau_i}{\tau_j}\right)}{l_j - l_i} \cdot \frac{\cos\alpha_0}{\cos\alpha_i}|\bar{r}_0|\hat{r}_i \tag{B.41}$$

$$= \frac{\omega_{i0}}{\omega_{j0}} \cdot \frac{\cos\alpha_0}{\cos\alpha_i}|\bar{r}_0|\hat{r}_i \tag{B.42}$$

$$\bar{r}_i = \frac{\omega_{i0}}{\omega_{j0}} \cdot \frac{\cos\alpha_0}{\cos\alpha_i}|\bar{r}_0|\hat{r}_{sc-l_i} \tag{B.43}$$

recall that $\bar{r}_0$ 0 is the vector from the spacecraft to the ejection location, $\bar{r}_{sc-l_i}$ is the vector from the spacecraft to the observation l_i, and ω_{ij} j is the observation angle rate computed using the ith and jth observations (or the radiant for index 0).

List of Figures

List of Tables

List of Symbols

Variable	Description	SI unit
$\boldsymbol{u}$	solid displacement	m
$\boldsymbol{u}_f$	fluid displacement	m

[1]Where the phase function is defined with the exponent given in the table, plus or minus the associated uncertainty

[2]Masses are expressed by means of a power law as $y = kx^{\alpha}$ where k and *alpha* are constants, and *alpha* is known as the *power-law exponent*. The notation used in the table distinguish in best estimate, upper and lower bound.

They ascended five or six more steps, and then Dantès felt that they took him, one by the head and the other by the heels, and swung him to and fro. ‘One!’ said the grave-diggers, ‘Two! Three!’ And at that instant Dantès felt himself flung into the air like a wounded bird, falling, falling, with a rapidity that made his blood curdle. At last, with a horrible splash, he darted like an arrow into the ice-cold water. He had been flung into the sea...
Alexandre Dumas, Le Comte de Monte-Cristo, 1844