PUZZLE SOFTWARE FOR THE CHARACTERISATION OF IN-ORBIT FRAGMENTATIONS

Matteo Romano[(1)], Andrea Muciaccia[(1)], Mirko Trisolini[(1)], Pierluigi Di Lizia[(1)], Camilla Colombo[(1)], Alessandra Di Cecco[(2)], and Luca Salotti[(2)]

[(1)]*Politecnico di Milano, Via Giuseppe La Masa 34, 20156, Milano, Italy, Email:*
{matteo1.romano,andrea.muciaccia,mirko.trisolini,pierluigi.dilizia,camilla.colombo}@polimi.it
[(2)]*Agenzia Spaziale Italiana (ASI), Via del Politecnico snc, 00133, Rome, Italy, Email:*
{alessandra.dicecco,luca.salotti}@asi.it

ABSTRACT

In recent years, fragmentation events have become more frequent and more difficult to predict and avoid, with growing risk for the safety of space operations. The constant surveillance and tracking of space objects facilitates the prediction of such events and allows to back track new objects and identify whether they originated in such an event. The PUZZLE software package was developed at Politecnico di Milano, under a contract with the Italian Space Agency (ASI), with two main objectives: identifying which unclassified debris, originated via a collision or explosion, and characterising the event in terms of mass and energy involved. The proposed approach focuses on the evolution of the osculating orbital elements of a large set of objects to identify common aspects of their motion, with pruning and clustering algorithms used to identify the epoch of a fragmentation and which known objects were involved.

Keywords: In-orbit fragmentations; Space debris characterisation.

1. INTRODUCTION

Space debris have become a persistent problem and a growing concern for operations of satellites orbiting around the Earth, with several fragmentation events occurring every year. The increasing number of launches and operative satellites leads to the production of more debris, as the probability of collisions increases as well [1]. Being able to predict and avoid possible explosion or collisions in orbit involving operating spacecraft or known debris is key to ensure safe and continuous operations of satellites providing a service. However, not all events can be predicted (e.g. explosions of rocket bodies or dismissed satellites) or avoided (e.g. collisions between objects in orbit). Debris produced by these events must be identified as soon as possible upon their occurrence, to detect fragmentations and reduce the risk they pose for other satellites in the future in a reliable and efficient manner.

Previous works aimed at fragmentation detection focused on different traits of the orbital motion of the fragments in order to estimate the time and place of the event and identify the parent object(s).

In the work of Andrisan et al. [2], the Simulation of On-Orbit Fragmentation Tool (SOFT) is used to characterise a recent fragmentation based on the detection of new debris by the Space Surveillance and Tracking (SST) network. The type of fragmentation (collision or explosion) is first determined based on the number of debris detected and the spreading of their orbital elements (both sensibly larger in the case of a collision). The time and place of the event are determined based on the average distance between the objects in the debris cloud, while the position of their centre of mass is used to identify the parent object(s). The comparison between the observed fragments and their modelling via a breakup model is used to remove unrelated objects from the set.

Frey et al. [3] [4] proposed a method to detect past fragmentations over long periods of time (of the order of years) exploiting the convergence of mean Keplerian elements when propagating backwards in time. The method analyses the clustering of inclination and right ascension of the ascending node (RAAN) in LEO to determine the epoch of the fragmentation as the values of the two parameters focus to similar values, as the objects involved since debris are likely to present similar orbital planes in the proximity of the fragmentation. The propagation is carried out with a semi-analytical method and a continuum formulation, which models the density of objects, rather than the objects separately, as function of time and orbital elements and other parameters (e.g. area-to-mass ratio).

Dimare et al. [5], instead, defines a similarity function between the orbital elements of the objects under examination to establish a metric for the identification of the fragmentation. Under the assumption that a fragmentation has occurred, the epoch of the event and the parent

Proc. 8th European Conference on Space Debris (virtual), Darmstadt, Germany, 20–23 April 2021, published by the ESA Space Debris Office
Ed. T. Flohrer, S. Lemmens & F. Schmitz, (http://conference.sdo.esoc.esa.int, May 2021)

object(s) are found by locating the minimum of the similarity function among the various objects, both on the short and the long term. Different metrics are analysed: the D-criterion proposed by Southworth and Hawkins [6], the Minimum Orbital Intersection Distance (MOID), and the nodal distance, with the first one found to be the most suitable for the problem.

The PUZZLE software package was developed at Politecnico di Milano with two main objectives: first, identifying which debris, inside a set of unidentified objects, originated via a collision or explosion; second, characterising the event (if any has occurred) in terms of mass and energy, identifying which known objects were involved, and modelling the distribution of the generated debris cloud over the space of orbital parameters and of physical characteristics (e.g. the area-to-mass ratio).

The approach proposed in this work achieves the two goals by analysing a set of unclassified objects in the form of Two-Line Element (TLE) data taken from a daily updated catalogue. Contrary to the methods introduced above, PUZZLE does not assume a new fragmentation has occurred recently, rather tries to determine whether it happened by trying to discern the known satellites waiting to be recognised from the newly formed debris.

The objects are propagated backwards to analyse their evolution in time, searching for a convergence of their osculating orbital elements. Pruning and clustering algorithms are employed to identify possible orbital intersection windows that make close encounters between objects possible, removing unrelated ones and identifying those which will appear in a small region of space at the same time. These are then matched with a catalogue of known objects to provide a guess of the possible operative or dismissed spacecraft involved in the explosion or collision. The fragmentation is then modelled using the available NASA standard breakup model, which provides distributions of area-to-mass ratio and relative velocity of the fragments useful to identify the orbital regions at risk of possible collisions in the future.

The tool described above is developed under a contract with the Italian Space Agency (ASI) as part of a more general software for the support of SST services and the study of space debris. In Section 2 its general architecture will be explained in detail with attention to the operations performed within each module. Then, in Section 3, the application of the software to actual fragmentation events will be shown alongside numerical results and performance data.

2. SOFTWARE ARCHITECTURE

This section describes the general architecture of the PUZZLE software.

The main goal of PUZZLE is the characterisation of in-orbit fragmentations. The software analyses a set of TLE data taken from a catalogue updated daily searching for possible fragmentations occurred in the near past, on time scales of the order of a few days, by associating the estimations of the orbital states of the objects to fragmentation events. While the completion of this task is severely dependent from the detection of new fragments and the presence of their TLEs among the published objects, the successful detection of such an event and of the objects involved in it allows the estimation of the masses and the energy levels characterising the fragmentation.

Together with the TLE data, the main inputs of the software are a series of parameters that are used in the algorithms explained in the following sections in order to detect the presence of a fragmentation in the days prior to the start of the analysis. These parameters include, but are not limited to:

- the length of the interval in which the search is done, of the order of a few days;
- the distance and time thresholds used to detect a close encounter between two objects;
- the parameters used to define whether two objects share an origin or not;
- the parameters useful to model the fragmentation (if any is found) and estimate the the number and characteristics of the fragments likely generated in the event.

Regarding the last point, the operation of modelling the breakup of the objects involved in the fragmentation is a necessary task for two reasons: on the one hand, since the detection, tracking, and classification of fragments happens over the days following the even, it facilitates the detection of the largest fragments as the predicted orbit can be scanned with prior knowledge; on the other hand, it allows to estimate which regions of space are at risk of collisions between known objects (operative satellites and debris) and the fragments with dimensions below a few centimetres, as these represent the largest portion of fragments usually generated in such events, but are also impossible to detect with classical means.

The software is composed of a series of five distinct modules, each performing a specific task, as represented in Figure 1:

Figure 1. Block diagram of the software architecture.

Module 1: reading and pre-filtering

The first step performed to analyse the input data is a pre-processing aimed at removing from the pool possible TLEs with erroneous values due to errors in the initial orbit determination process. This step ensures that the accuracy of the successive phases of the analysis is not degraded by non-coherent data that would in turn decrease the accuracy of the results. A diagram of this first block of operations is shown in Figure 2.

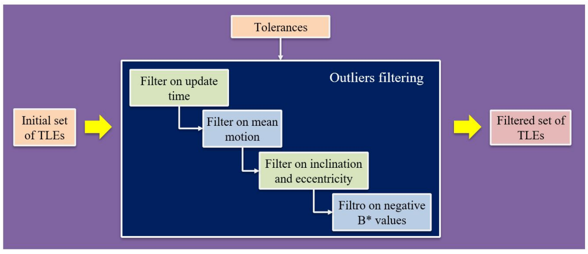

Figure 2. Block 1: module containing the routines for reading and pre-filtering the initial data.

The module reads the TLE data provided as input, selecting the ones corresponding to objects that are yet to be catalogued. Then, for each series of TLEs corresponding to the same Satellite Catalogue Number[1], a pre-filtering is performed to remove those TLEs whose values are statistical outliers. The filtering algorithm is the one proposed by Lidtke et al. [7], which is composed itself of five successive steps:

1. removing the TLEs corresponding to a correction of the immediately previous element, according to a minimum threshold of the update time between two subsequent TLEs;
2. identifying large gaps between TLEs to define time windows in which outliers will be searched (as shown in Figure 3);
3. removing the TLEs with values of mean motion that are not coherent within the same temporal window, using a sliding window approach (as shown in Figure 3);
4. removing the TLEs with values of inclination that are not coherent;
5. removing the TLEs with values of eccentricity that are not coherent;
6. removing the TLEs with negative values of B* drag term.

Outliers in mean motion, and the related semi-major axis, are used to detect propulsive manoeuvres, which result in a sudden change of the rate of change of mean motion $\dot{n}$, and thus in orbital energy.

Single outliers are detected by sliding a window, containing a fixed number of TLEs, through the TLE time series. A polynomial $n_{REG}(t)$ of a given order is obtained via regression of the TLEs contained in the window, and then compared against the first TLE value in the following window, as shown in Figure 3.

As shown in Figure 4, the predicted change in the mean motion $\Delta_P = n_{REG}(t_{i+1}) - n_{REG}(t_i)$ is defined as the difference in mean motion according to the regression polynomial between time t_{i+1} (corresponding to the last TLE in the sliding window) and time t_i (corresponding to the first TLE following the window). Similarly, the residual $\Delta_A = n_{REG}(t_{i+1}) - n_(t_{i+1})$ is the defined as the difference between the value of mean motion predicted at time t_{i+1} via the regression and the value contained in the TLE at that time. Δ_P and Δ_A are used to define the relative threshold T_R used to detect outliers based on the differences in mean motion of the subsequent TLEs:

$$T_R = \frac{\Delta_A}{\Delta_P} \tag{1}$$

Since Δ_P may be small if the mean motion does not change significantly inside the siding window, T_R may grow beyond the allowed tolerance. To avoid the detection of false positives, an absolute threshold T_A on the value of mean motion is defined, as

$$T_A = \frac{\Delta_A}{n_{REG}(t_i)} \tag{2}$$

The thresholds defined above are recomputed as the window slides, in order to account for the natural evolution of the mean motion of the orbit due to perturbations.

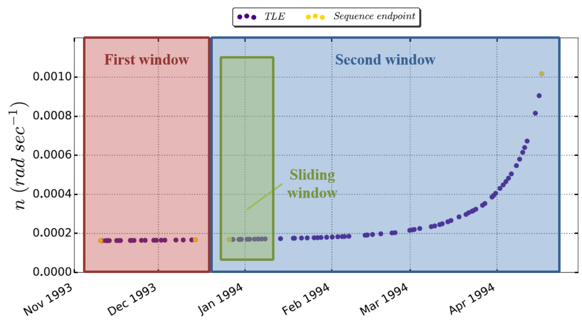

Figure 3. Division of a series of TLE in large windows (red and blue) based on the values of mean motion, and sliding window for the search of outliers (green). Image modified from [7].

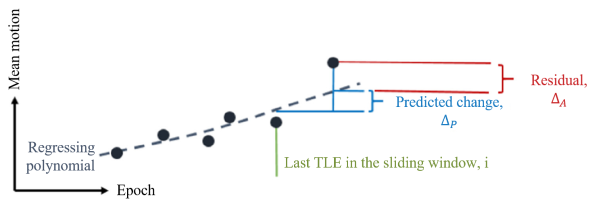

Figure 4. Regression used to identify outliers of mean motion in each sliding window. Image modified from [7].

[1]The classification followed in this work, as well as the source for all the TLE data, is the one of *space-track*.

Similarly, outliers in inclination and eccentricity (measured using the perigee radius r_P) are also searched for using the sliding window technique. However, in the case of these parameters, a simpler filtering technique is used. A window containing a fixed number of TLEs length is slid through the time series of each orbital element, and the median value in the window is computed. This median value is subtracted from the orbital element value of the TLE in the middle of the window, thus obtaining a time series of differences. Another window is slid through the time series of differences, using the Mean Absolute dDviation (MAD) to quantify dispersion. TLEs whose inclination and perigee radius differences are above a given MAD threshold are filtered out.

Finally, TLEs with negative values of the B* drag term are removed. This choice is done as negative B* values are likely results of modelling errors and uncertainties and would reduce the accuracy of orbital propagation [8].

The tolerances used in these operations are selected by the user prior the analysis. For a more detailed explanation of the algorithms, the reader is referred to the reference [7].

Module 2: pruning and clustering

In the second step of the fragmentation analysis, orbital intersections are searched for within the specified temporal window to remove those TLEs corresponding to objects that cannot have close encounters with each other and help identifying those that might share a common origin. The triple-loop filter proposed by Hoots et al. [9] is used for this purpose, as represented in the block diagram in Figure 5.

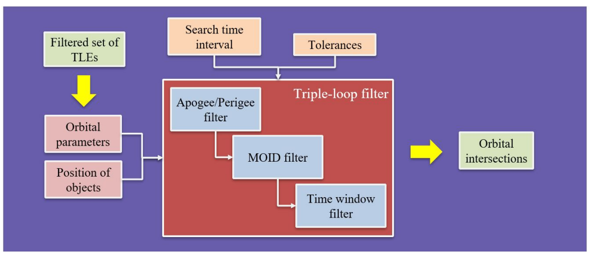

Figure 5. Block 2: module containing the routines for the pruning and the formation of families among the objects being analysed.

As per its name, the triple-loop consists of three filters working in series, two geometrical ones and a time-based one, which are applied to compare each couple of orbits contained in the TLE set after the pre-filtering phase.

The first geometrical filter compares the heights of the apogee and perigee of the two orbits to estimate if their geometry allows close approaches between the two objects; the quantities $q = max(r_{P,1}, r_{P,2})$ and $Q = min(r_{A,1}, r_{A,2})$ are defined, as shown in Figure 6, and then compared against a given threshold to defined whether the two orbits pass the filter:

$$q - Q \leq \Delta \tag{3}$$

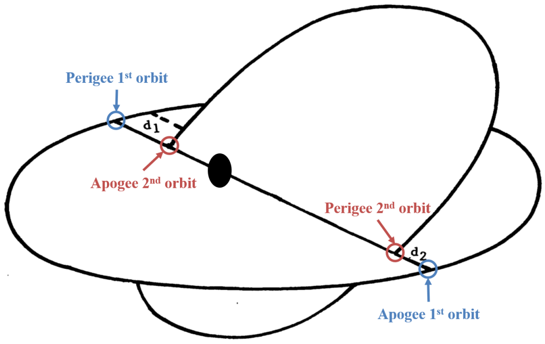

Figure 6. First step of the triple-loop filter: comparison between perigees and apogees of the two orbits. Image modified from [9].

The second geometrical filter evaluates the MOID between the two orbits to check whether it is below a given threshold. The MOID is computed here using the algebraic method proposed by Gronchi [10], which identifies the 16 closest geometrical points between the two selected orbits, among which the minimum value is selected for the comparison. While the MOID does not correspond to the actual minimum distance reached by the two objects moving along the orbits, it is the minimum possible distance between the two orbits; thus, when it is too large, no close encounter can occur between the objects.

The third and last filter defines angular windows around the the positions of the MOID along the two orbits to check whether it is possible for the two objects to be in the windows at the same time within a selected time period. Figure 7 shows how the windows are defined.

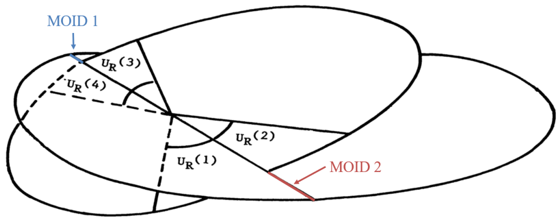

Figure 7. Third step of the triple-loop filter: definition of the angular windows around the MOID. Image modified from [9].

In the figure, an aperture angle u_R is defined around the position of the MOID between the orbits and its opposite point along the nodal axis. The angular windows are converted to time windows using Kepler's equation, and, by adding multiples of the periods of both orbits to the endpoints of each window, a sequence of time windows are defined throughout the interval set for the search of the fragmentation. The windows are then cross matched for possible overlaps: if at least two intervals along the orbits overlap, the two objects are able to experience a close encounter within the specified time frame; otherwise, no close encounter is possible as their orbital motion is out of phase.

If both objects satisfy the three filters, a close encounter is possible within the search time interval; otherwise, the TLEs representing them are removed from the set.

While until this point the three filters were used to prune the set of orbits (and their corresponding TLEs), filtering out those objects incapable of having close encounters with each other, the algorithm now computes the actual encounter distance. For each overlapping time window, a candidate time is calculated as the midpoint of the overlap. These candidate times are used as starting values for an iterative solution of the time of closest approach.

This is found by computing the time when the minimum distance between the two objects is reached. With reference to Figure 8, $\mathbf{r}_1(t)$ and $\mathbf{r}_2(t)$ are defined as the position vectors of the two objects at time t, $\mathbf{v}_1(t)$ and $\mathbf{v}_2(t)$ as the velocity vectors, and $\mathbf{a}_1(t)$ and $\mathbf{a}_2(t)$ as the accelerations acting on the two. The square of the relative distance between the two objects is then defined, using the rule of cosines, as:

$$\begin{aligned} d^2(t) &= r_1(t)^2 + r_2(t)^2 - 2\left(\mathbf{r}_1(t)\cdot\mathbf{r}_2(t)\right) \\ &= \left(\mathbf{r}_1(t)\cdot\mathbf{r}_1(t)\right) + \left(\mathbf{r}_2(t)\cdot\mathbf{r}_2(t)\right) - 2\left(\mathbf{r}_1(t)\cdot\mathbf{r}_2(t)\right) \end{aligned} \tag{4}$$

To search for the minimum of $d^2(t)$ means to search for the zeros of its derivative R, defined as:

$$\begin{aligned} R &= \frac{\mathrm{d}d^2}{\mathrm{d}t} \\ &= (\mathbf{r}_1\cdot\mathbf{v}_1) + (\mathbf{r}_2\cdot\mathbf{v}_2) - (\mathbf{v}_1\cdot\mathbf{r}_2) - (\mathbf{r}_1\cdot\mathbf{v}_2) \end{aligned} \tag{5}$$

where the dependence on the time t is left implicit for the sake of simplicity.

Newton's iterations are used to search for the zeros of the function R defined in (5):

$$t_{i+1} = t_i - \frac{R}{\dot{R}} \tag{6}$$

where

$$\begin{aligned} \dot{R} = v_1^2 &+ (\mathbf{r}_1\cdot\mathbf{a}_1) + v_2^2 + (\mathbf{r}_2\cdot\mathbf{a}_2) \\ &- (\mathbf{a}_1\cdot\mathbf{r}_2) - 2\left(\mathbf{v}_1\cdot\mathbf{v}_2\right) - (\mathbf{r}_1\cdot\mathbf{a}_2) \end{aligned} \tag{7}$$

It is to be noted that the implementation of the triple-loop filter used here does not account for orbital perturbations, and uses orbital parameters constant in time

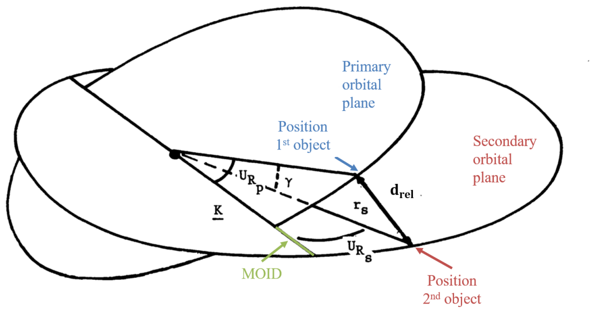

Figure 8. Computation of the global minimum encounter distance between a couple of objects inside each time window around the MOID between the orbits. Image modified from [9].

instead. However, orbital propagation in the following phase of the analysis allows to include the effects of the natural evolution of the orbits on the parameters used to search for orbital intersections, as will be explained in the next section.

The tolerances and thresholds (such as the distance margins between the orbits) used in these operations are defined by the user. For a detailed explanation of the algorithm, the reader is referred to the reference [9].

Module 3: propagation

In the third phase of the analysis, the TLEs that passed the previous filters are propagated backwards in time to identify possible convergences of the corresponding orbits. A diagram of these operations is shown in Figure 9.

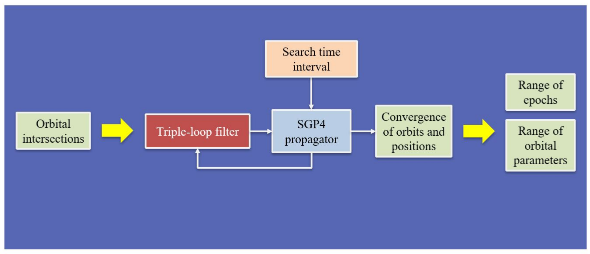

Figure 9. Block 3: module containing the routines for the propagation and the search for a convergence of the orbits being propagated.

The analytical SGP4 (Standard General Perturbations 4) model [11] [12] is used to propagate the sets of TLEs, since the information contained in a TLE is a set of averaged orbital elements that are specific to the SGP4 propagator. It considers secular and periodic variations due to Earth's oblateness, solar and lunar gravitational effects, gravitational resonance effects, and orbital decay using a drag model. The SGP4 propagator generates ephemeris in the True Equator Mean Equinox (TEME) coordinate system based on the epoch of the specified TLE. Due to the simplifications introduced by the analytical modelling of the perturbations, the accuracy of the propagation is generally limited to intervals of the order of a few days [13]. For this reason, the software limits the search for possible fragmentations to a maximum of 14 days. Future work will focus on the extension of the fragmentation search to longer time scales (of the order of months or years) using a semi-analytical formulation and averaged orbital elements.

In this phase, the triple-loop filter is used to identify the various windows where an intersection between any couple of orbits is possible, using the criteria explained in the previous section. In this case, the effects of perturbations on the evolution of the orbits are taken into account by the propagator. Once possible encounter windows and the corresponding minimum distances are estimated using the Hoots algorithm, each couple of object is propagated to the first time of closest approach to re-estimate the encounter windows and approach distances between them; this is repeated until the end of the selected time frame, and the global minimum approach distance is computed.

As a result, each closest encounter between any couple of objects, with corresponding time and distance, is recorded as shown in Figure 10, in order to later identify a possible fragmentation based on the convergence of multiple objects backwards in time.

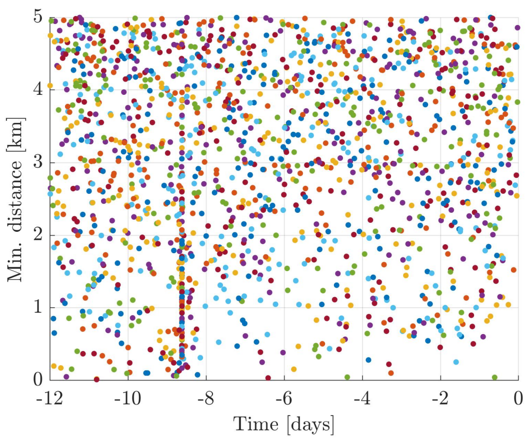

Figure 10. Distribution in time and corresponding distances of the closest approach for each couple of objects in the TLE set capable of experiencing a close encounter within the selected time frame.

The tolerances and other parameters (such as the length of the search period and the margins defining a close encounter) used in these operations are selected by the user.

Module 4: fragmentation search

In the fourth fase, the possible fragmentation is detected and the possible objects involved in it (parents and fragments) are identified. A scheme of it represented in Figure 11.

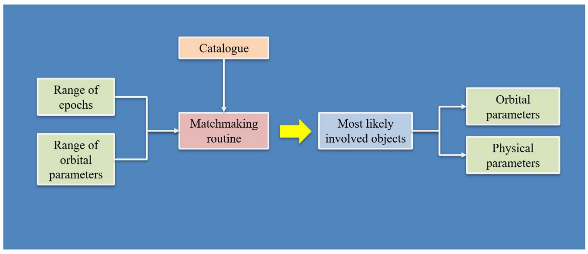

Figure 11. Block 4: module containing the routines for the identification of the fragmentation epoch and of the involved objects.

First, exploiting the data about close encounters between the objects gathered in the third phase (see Figure 10), a time window is identified around the possible epoch of the fragmentation event. This is done by dividing the interval selected for the analysis in bins and selecting the one presenting the most close encounters as the possible epoch.

Following this, the objects presenting close encounters in the so identified window are divided in groups based on their orbital parameters at the possible epoch of the fragmentation. This operation is done to refine the search for a convergence of objects among groups with similar orbital parameters, which are thus likely to have a common origin. The groups are defined following the single-linkage hierarchical clustering method initially proposed by Zappala et al. [14] for the definition of asteroid families.

In this method, a similarity distance function δv is assigned between a pair of objects as a metric to measure a separation between their coordinates in the space of orbital elements. Zappala et al. define the similarity distance using only the differences in semi-major axis a, eccentricity e, and inclination i of the orbits, and using proper elements to exclude the short-term evolution of the asteroid trajectories. However, here δv is defined using the osculating orbital elements of the objects contained in the TLE set (computed from the averaged orbital parameters using the SGP4 propagator), and including the RAAN Ω and the argument of periapsis (AoP) ω as well. This choice was done to account for the wider range of values of that these two parameters have when considering Earth orbiting obejcts compared to asteroids in deep space. The similarity distance is, thus, defined here as

$$\delta v = na \left[k_1 \left(\frac{\delta a}{a} \right)^2 + k_2 \left(\delta e \right)^2 + k_3 \left(\delta i \right)^2 + k_4 \left(\delta \Omega \right)^2 + k_5 \left(\delta \omega \right)^2 \right]^{1/2} \tag{8}$$

where n is the mean motion, and the k_i are weights associated to each difference of orbital parameters. Similarly to the definition proposed by Zappala et al., δv has the dimensions of a velocity increment, with the underlying idea that the similarity between two orbits is related to a deviation in velocity generated by disturbances.

The algorithm follows the steps described here:

1. given N objects of known coordinates, the similarity distance between each pair is computed;
2. the two closest objects, i and j, are identified and grouped together;
3. for any other object k, is the minimum between δv_{ik} and δv_{jk} is below a given threshold, the k object is added to the current group;
4. step 3 is repeated until no other object can be added to the current group;
5. a new group is defined, and steps 2-3 are repeated until all the initial N objects are assigned to a group.

The tolerances and other parameters used in these operations are selected by the user.

For each orbital group, the physical distance between each object in it is computed at the time identified as the possible epoch of the fragmentation. The objects presenting the lowest average distance are the ones selected as possible objects involved in the fragmentation, since their vicinity likely represent an epoch close to the actual fragmentation. The positions of the objects so identified are then compared with the ones of known objects (that is, with a Satellite Catalog Number and a Classification) to identify the possible parent objects and obtain their physical properties and orbital information.

Module 5: fragmentation modelling

The fifth and final phase of the analysis studies the distribution of the possible fragments generated in the event, starting from the information about the masses and positions of the parent objects gathered in the previous phase of the analysis. A scheme of the module is shown in Figure 12.

The current NASA Standard Breakup Model [15] [16] is used to characterise the fragmentation and to provide an estimation of the number of fragments formed in the fragmentation, as well of the distribution of their physical attributes (i.e. size, mass, relative velocity) based on the type of the fragmentation (whether a collision or an explosion), the type of object(s) involved (whether payload or rocket body), the total mass involved in the event, and, possibly, the collision speed. For a detailed explanation on how the statistical distributions are defined, the reader is referred to the references [15] [16].

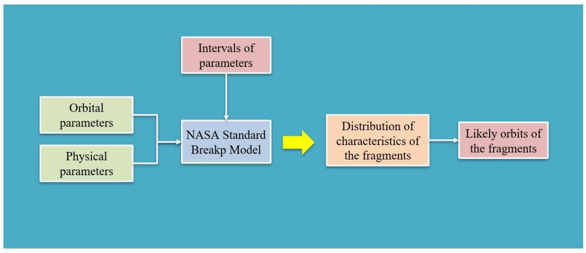

Figure 12. Block 5: module containing the routines for the modelling of the fragmentation using the NASA Standard Breakup Model.

In addition to the physical characteristics of the fragments, the distribution of relative velocity is combined with the information about the state of the parent object(s) at the epoch of the event to estimate the distribution of the orbital parameters of the fragment, in an attempt to identify the regions of space most affected by the fragmentation.

3. VALIDATION

The validation of the software is done by applying the search algorithms to two known fragmentation events:

- the collision the Iridium 33 and Cosmos 2251 communication satellites occurred on 10th February 2009;
- the explosion of the NOAA-16 meteorological satellite occurred on 25th November 2015.

In both cases, the TLEs of the fragments are taken from a daily catalogue dating back to some days after the event, and the analysis is performed on a set of TLEs including the ones of the fragments as well as the ones of random objects detected on that day. The accuracy of the analysis is evaluated by judging the correct identification of the fragmentation epoch, of the parent object(s), and of the number of involved objects that were included in the TLE set, while the efficiency is measured via the computational time. The main input conditions and results are reported in Table 1 and Table 2.

Considerations will be made in Section 3.3 on the sensitivity of the results accuracy and computational time from the main parameters used for the analysis.

3.1. Test case: Iridium-Cosmos collision

The initial set contains 2000 TLEs (23 of which referring to the two satellites and their 19 detected fragments) dating to 17th February 2009, 7 days after the event. The fragmentation is searched within the 10 days prior to the generation of the TLEs.

Figure 13, Figure 14, and Figure 15 show the orbits of the TLE set as it is processed during the analysis from the beginning, to the results of the triple-loop filtering, to the identification of those groups of objects with presenting close encounters in proximity of the possible epoch of the event, respectively. It is to be noted that, while Figure 13 shows only the LEO region, the initial set of 2000 TLEs actually contained objects in all kinds of orbits, ranging from LEO to GEO.

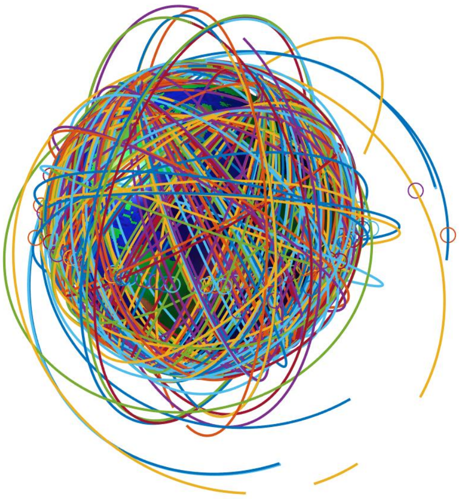

Figure 13. Orbits of the 2000 objects initially included in the TLE set for the Iridium-Cosmos test case, dating to 7 days after the collision (focus on the LEO region).

Figure 14. Orbits of the objects after passed the pre-filtering and the triple-loop filter for the Iridium-Cosmos test case.

Figure 17 and Figure 16 show how the fragments generated in the collision are distributed according to the NASA breakup model: Figure 17 shows the Gabbard diagram to highlight the change in orbital period and in the perigee and apogee of the orbits with respect to the orbits of the parent objects, while Figure 16 show the distribution of physical characteristics of the fragments.

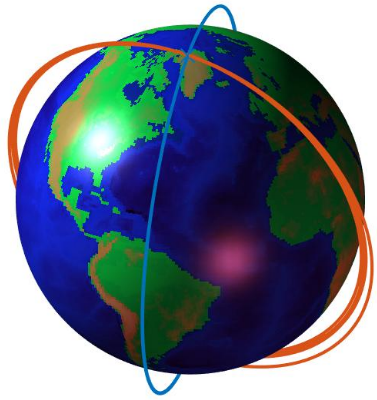

Figure 15. Orbits of the objects presenting close encounters in the interval around the possible epoch of the fragmentation, divided by group: the blue group is compatible with the orbit of Iridium 33, while the red group is compatible with the orbit of Cosmos 2251.

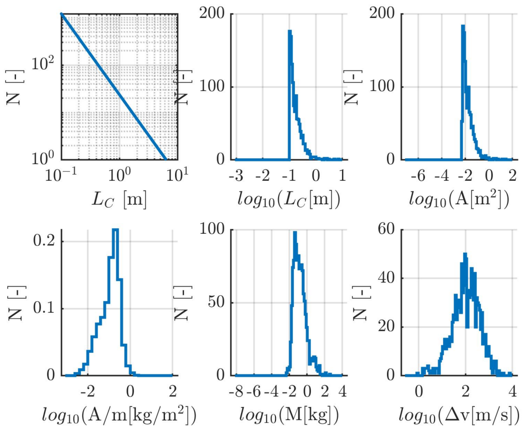

Figure 16. Gabbard diagram of the Iridium-Cosmos fragments generated via the breakup model, with the fragments from Iridium 33 in blue, the fragments from Cosmos 2251 in red.

Table 1 contains the main results of the analysis. The software was able to detect the fragmentation at the correct epoch, and to identify correctly the 19 objects involved in it (the 2 parent objects and the fragments) whose TLEs where present in the initial set. The computational time is of the order of a few minutes, due to the relatively low of objects non related to the fragmentation in the initial set of TLEs. More comments will be made about this and other aspects in Section **??**.

The high number of fragments estimated by the breakup model results from the type of fragmentation event: a catastrophic collision between satellites, with an impact speed of 11.647 km/s (estimated from the propagation of the TLEs of the two objects to the time of the event).

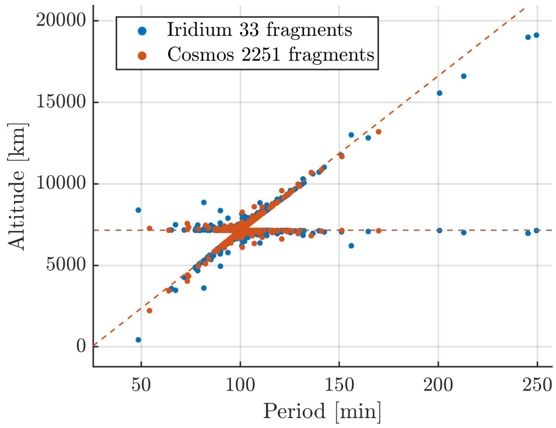

Figure 17. Distributions of the physical characteristics of the Iridium-Cosmos fragments generated via the breakup model. From top left to bottom right: cumulative distribution of characteristic lengths, distribution of characteristic length, of cross-sectional area, of area-to-mass ratio, of mass, and of relative velocity.

Table 1. Main parameters used to detect the Iridium-Cosmos fragmentation and main results.

Initial size of TLE set	~ 2000
Date of generation	17th February 2009
Time interval selected	10 days
Estimated epoch of the event	10th February 2009, 16:55:55
Number of objects involved	19
Probable parent object(s)	Iridium 33 (ID 24946), Cosmos 2251 (ID 22675)
Estimated number of fragments	1208 (367 from Iridium 33, 841 from Cosmos 2251)
Computational time	10.8 min

3.2. Test case: NOAA-16 explosion

The initial set contains 5000 TLEs (85 of which referring to the satellite and its 53 detected fragments) dating to 5th December 2015, 10 days after the event. The fragmentation is searched within the 10 days prior to the generation of the TLEs.

Figure 18, Figure 19, and Figure 20 show the orbits of the TLE set as it is processed during the analysis from the beginning, to the results of the triple-loop filtering, to the identification of those groups of objects with presenting close encounters in proximity of the possible epoch of the event, respectively. It is to be noted that, while Figure 18 shows only the LEO region, the initial set of 5000 TLEs actually contained objects in all kinds of orbits, ranging from LEO to GEO. Also, while Figure 20 shows that three orbit groups were identified in proximity of the possible event epoch, only one of them (plotted in blue) has objects with low enough relative distance between each other: these objects are the ones identified as the ones possibly involved in the event.

Figure 18. Orbits of the 5000 objects initially included in the TLE set for the NOAA-16 test case, dating to 10 days after the explosion (focus on the LEO region).

Figure 22 and Figure 21 show how the fragments generated in the collision are distributed according to the NASA breakup model: Figure 22 shows the Gabbard diagram to highlight the change in orbital period and in the perigee and apogee of the orbits with respect to the orbits of the parent objects, while Figure 21 show the distribution of physical characteristics of the fragments.

Table 2 summarises the main results for this test case. The software was able to detect the fragmentation at the correct epoch, and to identify correctly 23 out of 53 objects involved in it (the parent object and the fragments)

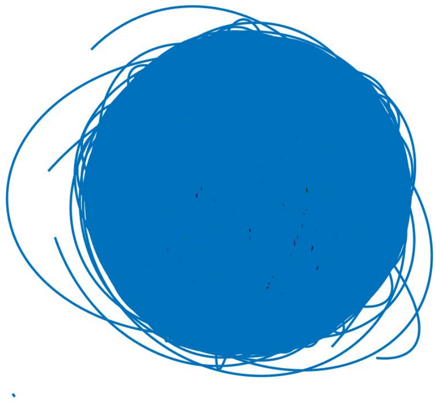

Figure 19. Orbits of the objects after passed the pre-filtering and the triple-loop filter for the NOAA-16 test case.

Figure 20. Orbits of the objects presenting close encounters in the interval around the possible epoch of the fragmentation, divided by group: the blue group is the most numerous and compatible with the orbit of NOAA-16.

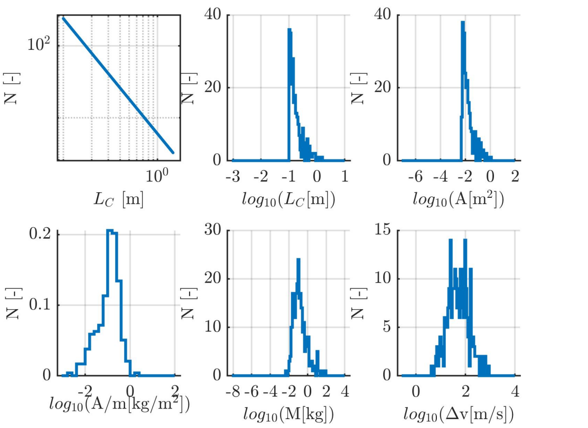

Figure 21. Gabbard diagram of the NOAA-16 fragments generated via the breakup model.

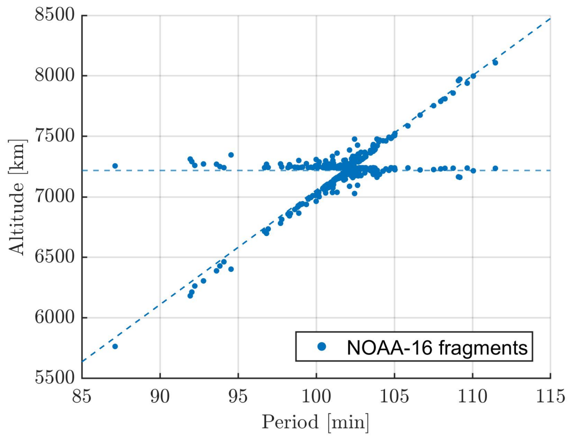

Figure 22. Distributions of the physical characteristics of the NOAA-16 fragments generated via the breakup model. From top left to bottom right: cumulative distribution of characteristic lengths, distribution of characteristic length, of cross-sectional area, of area-to-mass ratio, of mass, and of relative velocity.

whose TLEs where present in the initial set. The computational time is of the order of one hour, both due to the higher number of objects in the initial set of TLEs to be analysed with respect to the previous case, and due to the highr number of objects related to the fragmentation that increase the number of close encounter windows to be explored.

In this case, the number of estimated fragments is lower than the Iridium-Cosmos case due to the different ways in which the NASA breakup model simulates collisions and explosions, with more fragments being estimated in the first scenario, especially in the case of a catastrophic event. Similarly, while the NASA model generally gives large estimates of the relative speed of the fragments generated by the fragmentation, the distribution in the case of a collision tends to favour higher values, of the order up to 10^4 km/s, while it is more limited for explosions, as visible comparing the Δv distributions in Figure 16 and Figure 21. This causes the great variation in the orbital parameters of the fragments with respect to the parent objects seen by comparing the Gabbard diagrams in Figure 17 and Figure 22: while in the case of the explosion the variation is mostly limited to a few minutes and with few extreme values, in the case of the collision a larger dispersion and a higher number of extreme values can be observed.

3.3. Considerations on the input parameters

In this section, a few considerations are made about the sensitivity of PUZZLE to the parameters that are chosen to search for possible fragmentations. While some general remarks were already made in the previous section while commenting the validation results, specific considerations are presented here to identify which parameters

Table 2. Main parameters used to detect the NOAA-16 fragmentation and main results.

Initial size of TLE set	~ 5000
Date of generation	5th December 2015
Time interval selected	12 days
Estimated epoch of the event	25th November 2015, 07:16:44
Number of objects involved	23
Probable parent object(s)	NOAA-16 (ID 26536)
Estimated number of fragments	238
Computational time	57.3 min

affect the analysis the most in terms of accuracy and computational cost, focusing on specific cases and presenting numerical results to compare them. Thus, the Iridium-Cosmos test case is used here for these considerations.

In particular, the parameters that were selected for this analysis are:

- the number of TLEs included in the initial set;
- the number of actual debris produced in the fragmentation included in the initial set, which is analysed in Figure 23;
- the time past from the event to the generation of the initial TLE set, which is analysed in Figure 24;
- the distance margins used to detect a close approach (perigee/apogee distance and MOID distance) between any two objects during the search for orbital intersections, which is analysed in Figure 25.

The sensitivity of the analysis from these parameters is tested by comparing three performance and accuracy parameters: the computational time, indicated with (a) in the Figures; the error over the determination of the fragmentation epoch, indicated with (b); and the fraction of the objects actually involved in the fragmentation that were identified, indicated with (c).

Figure 23 shows the effect of the size of the initial TLE set and of the number of actual fragments included in it (ranging from 6 to the whole 23 available at the epoch of the TLEs) on the computational time (a), the determination of the event epoch (b), and the identification of the objects involved in the fragmentation (c). The TLEs were generated 7 days after the event, while the distance margins for the detection of close approaches were set to 5 km.

It is visible that the number of TLEs initially included in the set for the analysis is the main parameter affecting the computational time, due to the large number of possible intersection windows to be explored using the triple-loop. This phase accounts for the largest portion of the computational time, which grows to the order of hours when thousands of TLEs have to be pruned, while the pre-filtering phase only accounts for a very small fraction.

Similarly, the size of the initial TLE set also affects negatively the accuracy of the event epoch determination and of the identification of the objects involved in it, especially when very few TLEs belonging to the actual fragments produced in the event are included (red, purple, and yellow lines) against thousands of unrelated objects: this is due to the limitations of the way the epoch of the event is determined, based on the absolute number of close encounters between objects detected in a short interval of time. As the epoch is erroneously estimated, no objects compatible with a fragmentation around that epoch can be identified.

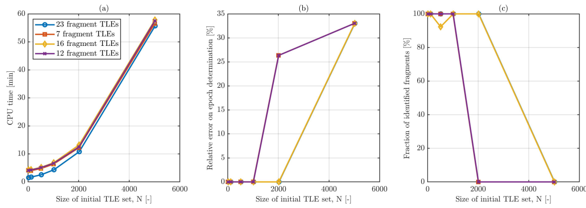

Figure 23. Sensitivity analysis of the computational time (a), the error on the determination of the event epoch (b), and the fraction of objects involved that were identified (c) against the number of TLEs included in the initial set, for different numbers of TLEs belonging to the actual fragments: 7 (red), 12 (purple), 16 (yellow), and 23 (blue).

Figure 24 shows the effect of the size of the initial TLE set and of the time past between the fragmentation and the generation of the TLEs (ranging from 7 to 14 days) on the computational time (a), the determination of the event epoch (b), and the identification of the objects involved in the fragmentation (c). 23 TLEs from the actual fragments were included in the initial set, while the distance margins for the detection of close approaches were set to 5 km.

In this case, as more time passes between the fragmentation and the generation of the TLEs being analysed, the effect on the computational time of the time past after the event is larger, with a doubling of it as the time interval doubles: this is due, again, to the higher number of intersection windows that are explored in search of close approaches between objects, due to the larger time interval where a fragmentation is searched for.

Similarly to the precious case, the time spent after the fragmentation also has a negative effect on the accuracy of the determination of the event, as the propagation with SGP4 becomes less reliable as larger time intervals are considered.

Finally, Figure 25 shows the effect of the size of the initial TLE set and of the distance margins used to detect close

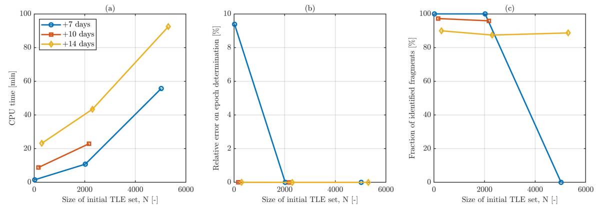

Figure 24. Sensitivity analysis of the computational time (a), the error on the determination of the event epoch (b), and the fraction of objects involved that were identified (c) against the number of TLEs included in the initial set, for values of the time past after the epoch of the fragmentation: 7 days (blue), 10 days (red), and 14 days (yellow).

approaches (ranging from 5 km to 20 km) on the computational time (a), the determination of the event epoch (b), and the identification of the objects involved in the fragmentation (c). 23 TLEs from the actual fragments were included in the initial set, and TLEs were generated 7 days after the event.

As in the previous two cases, considering larger margins to detect close encounters between objects increases the number of possible events that have to be estimated in search for the global minimum distance. In this case, however, this parameter has a smaller effect on the computational time than time past after the fragmentation, with similar effects on the accuracy of the results, where even a doubling of the margin leads to a failure in determining the epoch and the objects of the event.

However, despite not being shown, too low values of the distance margin may result in the missed detection of some close encounters and, thus, the missed detection of the fragmentation. This is due to the inherent inaccuracy of TLEs due to errors in orbit determination and of SGP4 approximated model, which introduce deviations in the states of objects of orders of hundreds of metres, thus affecting the encounter distance between them.

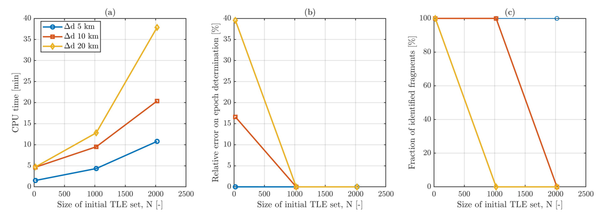

Figure 25. Sensitivity analysis of the computational time (a), the error on the determination of the event epoch (b), and the fraction of objects involved that were identified (c) against the number of TLEs included in the initial set, for different values of the distance margin used to detect close approaches between objects: 5 km (blue), 10 km (red), and 20 km (yellow).

To summarise the considerations done in this section:

- both the computational time and the accuracy of the results are improved when a smaller initial set of TLEs is considered, due to the lower number of objects to prune and lower number of orbital intersection windows to explore;
- the accuracy of the results is improved when a the initial set of TLEs contain a large portion of fragments actually belonging to the fragmentation, due to the higher number of close encounter they experience with each other which facilitates the detection of the event and the identification of the objects;
- both the computational time and the accuracy are improved when a short time has passed between the fragmentation event and the generation of the initial TLE set, due to the shorter time span to explore in search for a fragmentation and a higher accuracy of the propagation;
- both the computational time and the accuracy of the results are improved when shorter distance margins are used to detect close encounters between objects, due to the resulting lower number of intersection windows to explore; however, the accuracy might not benefit from margins that are too small, due to the propagation errors causing the objects to present encounter distances larger than actually occurred.

Thus, an ideal case for the application of PUZZLE would be represented by an initial set of TLEs composed mostly (if not only) by objects that were generated during the fragmentations and were detected immediately after the event. However, since the detection of the fragments generally takes days and only a few TLEs might be available shortly after such an event, and since many other unrelated objects are detected every day but only classified later, the occurrence of such conditions is almost impossible. Thus, more study of the effects of the various parameters used in the analysis is required.

4. CONCLUSIONS

Due to the increasing number of launches, the population of space debris surrounding the Earth is growing at an alarming rate, causing fragmentation events (collisions and explosions) to become more frequent and posing a threat to the safe operation of satellites. Thus, the early detection of these fragmentations becomes a key task in maintaining the safety of operative satellites.

The PUZZLE software was developed with the goal of characterising in-orbit fragmentations starting from a set of unclassified TLEs, with the goals of detecting whether a fragmentation has occurred in the recent past, determining which objects were involved, and estimating the distributions of characteristics of the fragments. This paper presented the architecture of PUZZLE, explaining in detail the operations performed within the five modules of the software and the algorithms used in them. Two test cases were presented and discussed, to show the capabilities of the software in detecting a fragmentation and the computational cost. Later, some considerations were made on how the choice of the parameters used for the analysis affect the accuracy of the results and the efficiency of the process.

The discussion presented here shows that PUZZLE is capable of detecting a fragmentation and correctly identify the epochs and the objects involved in most cases. This is achieved through use of algorithms which identify common aspects among the motion of the objects under examination, focusing on the evolution of the osculating orbital elements: the occurrence of a large number of close encounters between objects in a short time span via the triple-loop filter, and the similarity of the orbits of such objects via hierarchical clustering. In this way, the software can detect whether a set of objects converge to the same region of space going backwards in time, thus identifying a possible fragmentation event occurred at a specific epoch.

Limitations exist due to the sensitivity of the algorithms employed in the analysis from the size of the initial TLE set and the amount of fragments represented in it, and from the various parameters used to gradually prune the TLE set in search of a subset showing a behaviour compatible with a fragmentation. Some of them have been already identified (such as the distance margins to detect close encounters), however further study is required to identify other such influential parameters and, possibly, optimal values to ensure accurate and fast results.

Future work to the PUZZLE software will focus on improving some of the main algorithms used to detect fragmentations. The inclusion of perturbations in the short-term search for orbital intersection windows in the triple-loop filter will allow to explore said windows more efficiently, as fewer propagations will be needed to estimate the global minimum encounter distance between any two objects. The identification of the fragmentation epoch will be performed in a more efficient way, by considering variable time intervals in which the close encounters are clustered in order to avoid incorrect selection of an epoch based solely on the absolute number of close encounter in its proximity. The fragmentation search will be extended to longer time scales (of the order of months or years) using a semi-analytical formulation and averaged orbital elements. Finally, uncertainties over the states represented by the TLEs will be included, to allow for a better correlation between the orbital states of objects involved in the fragmentation, and between the unclassified objects and the known ones.

ACKNOWLEDGMENTS

The research activities described in this paper were performed within the European Commission Framework Programme H2020 “SST Space Surveillance and Tracking” contracts N. 785257 (2-3SST2016).

REFERENCES

1. McKnight D. S. (2016), ”Orbital debris hazard insights from spacecraft anomalies studies”, *Acta Astronautica*, Volume 126, 2016, Pages 27-34, ISSN 0094-5765, `https://doi.org/10.1016/j.actaastro.2016.04.005`
2. Andrisan R. L., Ionita A. G., González R. D., Sánchez Ortiz N., Krag H. (2017). Fragmentation Events analysis Making use of fragmentation model and Assessment Tool (FREMAT). 7th European Conference on Space Debris, Apr. 17-21 2017, Darmstadt, Germany
3. Frey S., Colombo C., Lemmens S. (2018). Advancement of a density-based debris fragment model and application to on-orbit break-up localisation. 36th IADC plenary meeting, Jun. 5-8 2018, Tsukuba, Japan.
4. Frey S., Colombo C., Lemmens S. (2019). Density based modelling and indication of break-up location and epoch from fragments using backwards propagation. 5th European Workshop on Space Debris Modelling and Remediation, Jun. 25-27 2018, Paris, France.
5. Dimare, L., Cicalò, S., Rossi, A., Alessi, E. M., Valsecchi, G. (2019). In-Orbit Fragmentation Characterization and Parent Bodies Identification by Means of Orbital Distances. 1st International Orbital Debris Conference (IOC), Dec. 9-12 2019, Sugarland, Houston, TX, USA.
6. Southworth, R.B. and Hawkins, G.S. Statistics of Meteor Streams, Smithsonian Contributions to Astrophysics, Vol. 7, pp. 261-285, 1963.
7. Lidtke, A., Gondelach, D., and Armellin, R. (2019). Optimising filtering of two-line element sets to increase re-entry prediction accuracy for GTO objects. Advances in Space Research, 63:1289–1317.
8. Vallado, D., Crawford, P., Hujsak, R., and Kelso, T. (2006). Revisiting spacetrack report no. 3: Rev. 2
9. Hoots, F. R., Crawford, L. L., and Roehrich, R. L. (1984). An analytic method to determine future close approaches between satellites. Celestial Mechanics, 33:143–158.
10. Gronchi, G. F. (2005). An algebraic method to compute the critical points of the distance function between two keplerian orbits. Celest. Mech. Dyn. Astron., 93:295-329, 09 2005.
11. Kelso, T., Hoots, F., and Roehrich, R. (1988). Spacetrack report no. 3-models for propagation of norad element sets. NASA, Tech. Rep.
12. Vallado, D., Crawford, P., Hujsak, R., and Kelso, T. (2006). Revisiting spacetrack report no. 3: Rev. 1
13. AGI (Ansys Company). Sgp4 propagator. `https://help.agi.com/stk/11.0.1/Content/stk/vehSat_orbitProp_msgp4.htm`.
14. Zappala, V., Cellino, A., Farinella, P., and Knezevic, Z. (1990). Asteroid families. i. identification by hierarchical clustering and reliability assessment. The Astronomical Journal, 100:2030–2046.

15. Johnson, N., Krisko, P., Liou, J.-C., and Anz-Meador, P. (2001). Nasa's new breakup model of evolve 4.0. Advances in Space Research, 28:1377–1384.

16. NASA (2011). Proper implementation of the 1998 nasa breakup model. Orbital Debris Quarterly News, 15(4):4–5.