Journal of Space Safety Engineering 11 (2024) 462–468

Contents lists available at ScienceDirect

Journal of Space Safety Engineering

journal homepage: www.elsevier.com/locate/jsse

AstroLibrary: A library for real-time conjunction assessment and optimal collision avoidance

Shawn SH Choi [a,b,§], Peter JH Ryu [a,b,§], Kyuil Sim [a,c], Jaedong Seong [d], Jae Wook Song [c], Misoon Mah [e], Douglas DS Kim [a,b,*]

[a] SpaceMap Inc., 222, Wangsimni-ro, Seongdong-gu, Seoul 04763, South Korea
[b] School of Mechanical Engineering/Voronoi Diagram Research Center, Hanyang University, Seoul 04763, South Korea
[c] Department of Industrial Engineering, Hanyang University, Seoul 04763, South Korea
[d] SSA Research Office, Korea Aerospace Research Institute (KARI), Daejeon 34133, South Korea
[e] M&K Research and Development Inc., Gainesville, VA 20155, United States

A R T I C L E I N F O

Article history:
Received 13 March 2024
Received in revised form 19 June 2024
Accepted 5 July 2024
Available online 12 July 2024

Keywords:
Conjunction assessment
Collision avoidance
Spatiotemporal problem
Real-time
Voronoi diagram

A B S T R A C T

Geospace is crowded due to the proliferation of satellites and space debris and will become more crowded with the increasing deployment of new space missions. This trend is rapidly increasing the probability of collisions between space objects. Space objects fly at extreme speeds; hence, the consequences of collisions are catastrophic. However, accurate and efficient conjunction assessment (CA) and collision avoidance (COLA) have long been challenging, even with the current space catalogues of O(10^4) size. As the space catalogue size increases owing to the increased number of new satellites, improved sensor capabilities, and Kessler syndrome, the situation will worsen unless a paradigm-transforming computational method is devised. Here, we present the SpaceMap method, which can perform real-time CA and near-real-time COLA for O(10^6) or more objects, provided that the spatiotemporal proximity amongst satellites is represented in a Voronoi diagram. As the most concise and efficient data structure for spatiotemporal reasoning amongst moving objects, Voronoi diagrams play a key role in the mathematical and computational basis for a new genre of artificial intelligence (AI) called space–time AI, which can find the best solutions to CA/COLA and other space decision-making problems in longer timeline windows. The algorithms are implemented in C++ and are available on GitHub as AstroLibrary, which has RESTful APIs and Python packages that can be called from application programs. Using this library, anyone with elementary programming skills can easily develop efficient applications for challenging spatiotemporal problems.

1. Introduction

Many objects, including uncrewed satellites, crewed space stations, and "space junk" are currently orbiting the Earth, with new and advanced objects being added every year. The plans of Starlink and Guowang constellations of O(10^4) satellites and the OneWeb and Kuiper constellations of O(10^3) satellites are being realized. A recent study on radio spectrum filings by the International Telecommunication Union [1] predicts that more than one million new satellites will be launched in the next few years. Space debris is another side of the space objects story. According to the European Space Agency (ESA) [2], O(10^4) space objects larger than 10 cm, O(10^6) objects larger than 1 cm, and O(10^8) objects larger than 1 mm orbit the Earth. NASA reported similar results [3]. The Kessler syndrome is also a key consideration [4]. However, only a small subset of space objects has been observed and catalogued. For example, the Space Track database included 44,700 objects as of October 2023 [5]. Space objects fly at extremely high speeds; hence, collision impacts can be catastrophic, even if the objects are small.

Hence, the prediction and avoidance of potential collisions between space assets have long been important research subjects. However, the available computing methods struggle to process the current catalogue of O(10^4) size for conjunction assessment (CA) and collision avoidance (COLA). The main limitation lies in the software rather than in the hardware. The principal conjunction algorithm is the 1984 three-filter algorithm [6], which divides the timeline into a set of mutually exclusive time segments and checks

* Corresponding author.
E-mail address: douglas.kim@spacemap42.com (D.D. Kim).
§ Equally contributed.

https://doi.org/10.1016/j.jsse.2024.07.003

the conjunction between each object pair in each time segment. Although this seminal algorithm has contributed profoundly to the field, it has a few fundamental drawbacks for the future space environment. It was designed to solve spatial problems between two objects at a time, and not amongst many objects. In addition, the algorithm is coordinate dependant. Furthermore, the memory requirement, which is inversely proportional to the length of the time step, can become relatively large. For example, in our experiment, more than 700 GB of memory was easily consumed with approximately 21,000 low-Earth orbit (LEO) objects in Space-Track for a 24-h window with a 1-s time step. Many prior studies have attempted to accelerate the three-filter algorithm using hashing mechanisms, such as 3D buckets, kd-trees, or sieves [7-9]. Another important perspective is that the computational results were not used for any purpose other than CA/COLA and were subsequently abandoned.

The space environment is rapidly changing from three seemingly conflicting perspectives: (i) an increase in the size of the space catalogue, (ii) the need for real-time CA/COLA, and (iii) the rise of new applications with more challenging computational requirements, such as optimization problems.

The space catalogue size is expected to surge, perhaps faster than exponential growth. In July 2023, the Intelligence Advanced Research Activity (IARPA) announced the Space Debris Identification and Tracking (SINTRA) program [10], which aims to detect, track, and characterize space debris as small as 1 mm. NASA issued a challenge, due in November 2023, for detecting, characterizing, tracking, and remediating debris of 1–10 cm in size [11]. In parallel, commercial space-based space situational awareness (SSA) efforts have been launched to detect millimetre- to centimetre-sized objects. We anticipate that these, and similar efforts to follow, will result in catalogues of immense size, which will make it difficult for existing algorithms to produce CA/COLA solutions at a reasonable speed.

Independent of the expected catalogue size, critical use cases add another dimension to the challenge. Human-on-board missions are the most critical use case because astronauts' safety must be guaranteed, for example through orbital tours. If the current transportation cost of O($1000)/kg to LEO becomes lower than O($100)/kg by, for example, Starship, both passenger and freight transportation between cities might be realized to supplement, if not replace, air transportation. For these use cases, **"fast computation"** is not sufficient: The **"real-time CA/COLA"** will be one of the determinants for the success of this transportation type; after updating catalogue itself in real-time new measurement data also obtained in real time.

We need to solve space decision-making problems in the following setting where $\otimes$ represents a Cartesian product:

$$\begin{aligned}&\textit{Immense catalogue} \otimes \textit{Real}-\textit{time CA/COLA}\\&\quad\otimes \textit{Real}-\textit{time data update} \otimes \textit{Optimization problems}.\end{aligned} \tag{1}$$

In this paper, we present a unified computational framework that performs real-time CA/COLA for catalogues of size O(10^6) or larger, with moderate computational resources. The spatial proximity amongst satellites is represented by a data structure called the Voronoi diagram, which is considered the most concise and efficient data structure for spatial reasoning amongst many objects. Voronoi diagrams play a key role in the mathematical and computational basis of a new genre of paradigms called Space–Time artificial intelligence (ST-AI), which can solve CA/COLA and other space problems in a longer timeline window. ST-AI is composed of both Voronoi diagrams and deep-learning methods.

We first preprocess space objects to produce important space-time events in a timeline and store them in the Space-Time Event List in Astro database (STELA DB). CA can then be obtained in real time by simply scanning the events in STELA. COLA can be solved in near real time by performing CA on multiple manoeuvre alternatives. Each alternative may result in secondary and tertiary conjunctions that are computationally demanding to resolve [12]. The CA of a trajectory through M waypoints is processed in O(M)-time, given STELA. Notably, this is independent of the catalogue size. If N cores are used for the computation, they can be answered in O(M/N) time. The proposed algorithm is highly scalable. A variation of the CA algorithm can be later used to marginally update the catalogue using SSA data collected in real time during flights. Diverse trajectory types, such as stationary orbits, launch trajectories, and orbit transfer trajectories, are handled homogeneously in our framework. The SpaceMap method also solves more challenging spatiotemporal optimization and intelligence problems with unparalleled efficiency. This approach can address NASA's concern regarding the heavy computation requirements of CA/COLA algorithms [12].

The proposed method is available from SpaceMap on AWS: spacemap42.com. The algorithms were implemented in C++ and are available as AstroLibrary on GitHub. This solution provides RESTful APIs and a Python package that can be embedded in application programs. The C++ version is also available. We developed it for all stakeholders to have easy access to our algorithms so that application programs for challenging spatiotemporal problems can be easily and conveniently developed by anyone with elementary programming skills. We believe that current extensive space investments will be sustainable only if they make profits and profits can be maximized with good software for both safety and optimization of space assets. We present the experimental results of some functions related to CA/COLA using LEO (low earth orbit) TLE (two-line element set) data downloaded from Space Track. Unless otherwise stated, the computational environment was as follows: OS: Ubuntu 20.04; CPU: 64 core / 128 thread (2.7 GHz); memory: 512 GB.

2. Related prior studies

Collisions in space are an important issue. In the intuitive and seminal 1984 three-filter algorithm [6], the first perigee–apogee filter quickly reduces the combinatorial search space by comparing the perigees and apogees of each pair of objects. The second path filter screens an object pair if the minimum distance between its orbital geometries exceeds a given threshold. The third filter divides the timeline into a set of short time intervals and solves the conjunction problem for each object pair (i.e., the distance problem between two objects). This algorithm has significantly contributed to the space community but has fundamental limitations for the changing space environment in the New Space Age.

Numerous studies have been conducted on CA since 1984. These are mostly related to the acceleration of the three-filter algorithm using geometric hashing with 3D buckets or kd-trees. The smart sieve developed in 2002 doubled the algorithm's efficiency by considering escape velocity [12]. However, the smart-sieve method adopted by most commercial systems inherits the limitations of the three-filter algorithm explained earlier. All acceleration schemes have attempted to reduce the number of object pairs in the search space.

In 2017, we introduced the Voronoi diagram approach to CA/COLA [13], followed by other improvements [14-15]. Voronoi diagrams are the most concise data structures for representing spatial relationships amongst particles, called generators, in space. These diagrams tessellate space into tiles, called Voronoi cells. The Voronoi cell is a set of locations closer to the corresponding generator than any other location. It is particularly well-suited for efficiently solving spatial reasoning problems amongst generators in 2D and 3D space. Spatiotemporal (i.e., space–time) reasoning problems amongst moving generators are solved by first constructing a dynamic Voronoi diagram (DVD) of the moving generators and taking advantage of the spatial reasoning power of the Voronoi diagrams at distinct moments. For more details, refer to [16].

Regardless of the algorithm for finding the times and distances of the closest approaches, the source and quality of input data matter, i.e., the quality of the state vectors of resident space objects. There are several space situational awareness (SSA) databases: TLE of Space-Track, the high accuracy catalogue (HAC) of Space-Track, commercial databases such as LeoLabs, Slingshot, ExoAnalytic, Safran, etc. Many space-based SSA databases will be available in near future as well. Despite the consensus of relatively low accuracy, the practical value of TLE persistently remains as witnessed by efforts such as Euro Space Agency (ESA) [17] and Indian Space Research Organisation (ISRO) [18]. Both studies used TLE as the input data to predict conjunctions. In addition, there are efforts to improve the usefulness of TLE continue [19-21]. While our study here is based on the Space-Track TLE database, AstroLibrary can read in any SSA database following a standard ephemeris format without any problem. A comparison of the TLE data against other SSA databases will be an important future research work.

3. Catalogue conjunctions

Catalogue conjunction is a conjunction of catalogue objects. Astro-1 can solve the CA of a particular catalogue object or all catalogued objects in a few milliseconds using the STELA DB. We benchmarked the quality of the conjunction solutions against Celestrak [22]. Astro-1 synchronizes the TLE download time with Celestrak three times daily, at 0800, 1600, and 2400 UTC.

Fig. 1 shows the correlation of the CA solutions produced by Astro-1 and Celestrak for the Top-10, Top-100, Top-1000, and Top-10,000 conjunctions. Fig. 1(a)–(d) shows the distances of closest approach (DCAs). As indicated by the red diagonal, Astro-1 tends to be slightly more conservative than Celestrak, although their correlation is high. Both programs produced identical conjunction sets of catalogue object pairs. Repeated experiments produced identical results. We also compared the Astro-1 solutions with those produced by STK to reach the same conclusions. Fig. 1(e)–(h) shows the difference in times of closest approach (TCAs) between Astro-1 and Celestrak, where the average is close to zero with small deviations. Fig. 1(i)–(l) shows the probability of collisions on a log scale with base 10. Astro-1 uses the maximum collision probability formula proposed by Alfano [23,24], similar to Celestrak:

$$P_{c\ \max}\left(\sigma_x^2\right) = \exp\left(-\frac{x_e^2}{2\sigma_x^2}\right)\left[1 - \exp\left(-\frac{\alpha r_A^2}{2\sigma_x^2}\right)\right] = f(r_A,\ x_e) \quad (2)$$

where α is the aspect ratio (AR), given as $AR = \sigma_x/\sigma_z = 3$; r_A is the hard-body radius, HBR; x_e is the DCA; and σ_x^2 is the error variance, given as $\sigma_x = x_e/\sqrt{2} = DCA/\sqrt{2}$. Hence, Pc is a function of the HBR and DCA. Astro-1 and Celestrak have good agreement in their DCAs; hence, the difference in the probability profile lies in the HBR estimation by the different methods. We retrieve the geometric attributes of both the primary and secondary objects from the ESA database and calculate the minimal sphere enclosing the objects. We then define HBR by adding the radii of the two enclosing spheres. If the geometric attributes are not available in the ESA database, a default radius of 1 m is assigned. The access to the equivalent NASA database is desirable.

4. Phantom conjunctions

A phantom is a spatiotemporal object that is not in the catalogue. Correspondingly, a phantom conjunction is a conjunction between a phantom and an object in the catalogue. A phantom may be a new satellite that its operators plan to insert into orbit, a space plane that flies in inter-orbital space, or a hypothetical object. It propagates together with the catalogue objects and is associated with its trajectory in the form of an ephemeris or TLE.

Efficient assessment of phantom conjunctions has many critical applications. Examples include the optimal design of constellations for carrying out missions safely and efficiently and constellation evaluation for insurance companies.

The CA of a phantom can also be efficiently performed using the STELA DB. The basic idea of the phantom CA algorithm is to perform a spatiotemporal neighbourhood search in the Voronoi diagram in STELA and perform CA. The key determinant is reducing the solution space to the extent possible while ensuring that no solution is missing.

Fig. 2(a) shows the computation time for phantom conjunctions by Astro-1 using 100 phantoms for a 24-h window. Old TLE data of 100 Starlink satellites was used to simulate phantoms. The top black lines show the computation time of the proposed code, and the solid and dotted lines correspond to the conjunctions computed within the thresholds of 100 km and 10 km, respectively. A 100-km threshold means that conjunctions within a 100-km distance are found. This difference is negligible. Note that approximately half of the processing time is used to upload the STELA DB to memory. The blue lines correspond to the scenario in which STELA resides in memory. We expect to achieve the bottom red line after the current code is reengineered. Our proposed algorithm is highly scalable; hence, doubling the computing power halves the computation time. Fig. 2(b) shows the number of conjunctions defined by a phantom with respect to the threshold distance over a 24-h window.

5. Optimal collision avoidance

We claim that Astro-1 can produce a COLA manoeuvre plan on a spacecraft, such as a satellite or space plane, better and faster than any other approach with the same catalogue. The idea is simple: generate-and-test. First, we generate a sufficient number of candidate trajectories and evaluate them quickly using the fast-phantom CA algorithm. In other words, each candidate trajectory is treated as a phantom. Subsequently, we choose the best one.

Let c(p, s, DCA, TCA) = conjunction(primary, secondary, DCA, TCA) be the conjunction between p and s with DCA and TCA, which we attempt to avoid. Let $W[t_s, t_e)$ be the time window used to evaluate the manoeuvreing trajectory quality. Note that TCA $\in$ W. Let TRJ = $\{trj_1, trj_2, .., trj_n\}$ be the set of n trajectories, where each is associated with a spatiotemporal chain, for example, TLE or ephemeris. For convenience of discussion, suppose that each trajectory is associated with a set of m time-stamped waypoints, e.g., $trj_i = \{(x_{i1},t_{i1}, \psi_{i1}), (x_{i2},t_{i2}, \psi_{i2}), .., (x_{im},t_{im}, \psi_{im})\}$, where x_{ij} is the spatial coordinate of the waypoint that the spacecraft passes through at the time t_{ij}. A waypoint is associated with a non-negative propellant consumption ψ_{ij}.

The quality of all elements in the TRJ set has to be evaluated. Suppose that W is 24 h and we want to start the manoeuvre in the middle of W. In other words, we want the manoeuvre to begin at TCA − 12 h and to evaluate the quality of the manoeuvre until TCA + 12 h. Hence, $W[t_s, t_e) = W$[TCA – 12 h, TCA + 12 h). Some waypoints in front of each trajectory are associated with positive propellant usage.

The trajectory quality can be defined in numerous ways.

Formulation 1. Maximum miss distance with propellant consumption constraint: Constrained integer linear programming problem

A simple yet effective measure of trajectory quality is the miss distance. Here, we aim to maximize the miss distance of the trajectory and formulate a max–min problem. Suppose that each trajectory corresponds to a phantom that follows that trajectory, and we

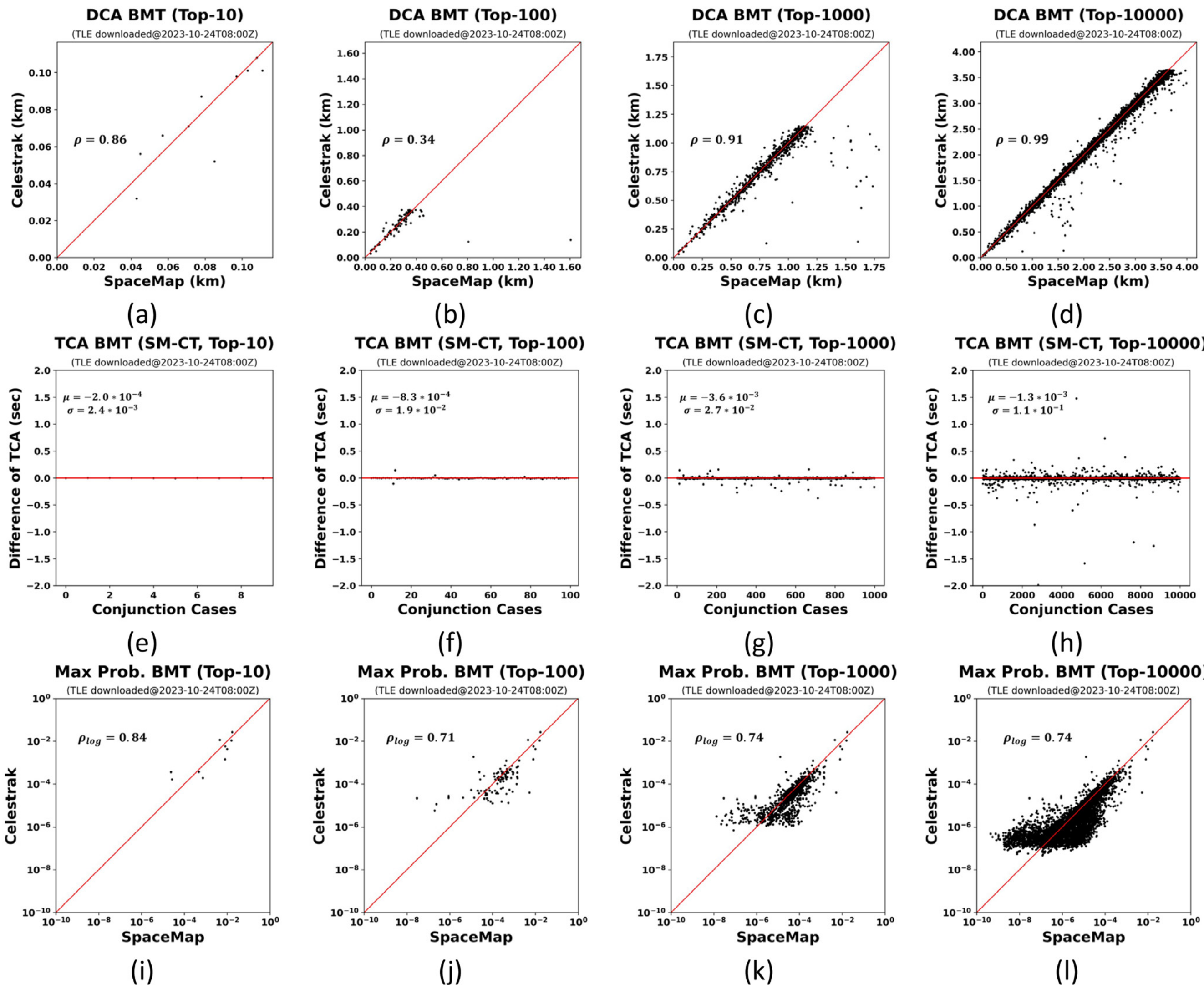

Fig. 1. Comparison of the CA solutions produced by SpaceMap's Astro-1 and Celestrak's Socrates Plus. TLE downloaded at 08:00, Oct. 24, 2023 (UTC). Celestrak conjunction data downloaded at 14:00, Oct. 24, 2023 (UTC). (a)–(d) Distances of closest approach. (e)–(h) Differences of the time of closest approach. (i)–(l) Probability of collision. (a), (e), (i) Top-10 conjunctions. (b), (f), (j) Top-100. (c), (g), (k) Top-1000. (d), (h), (l) Top-10,000.

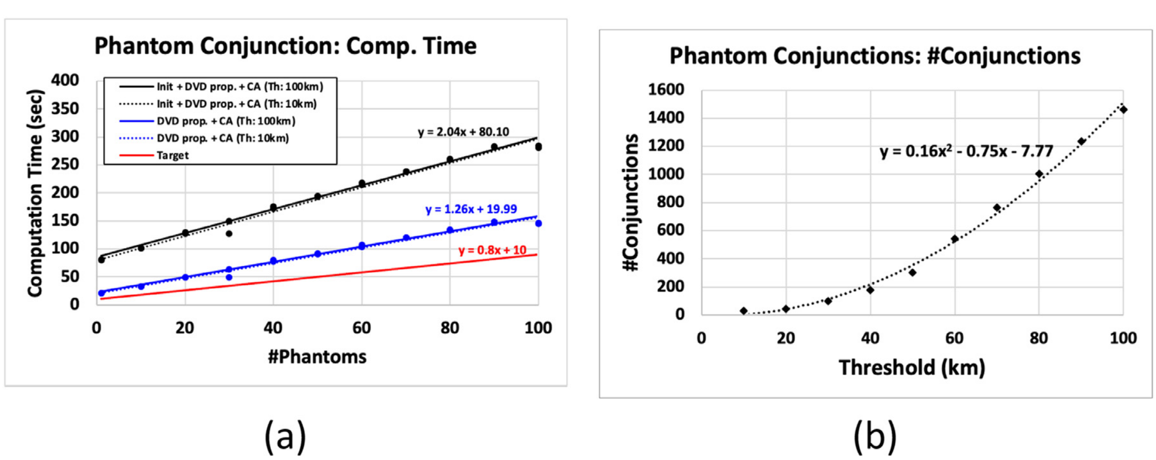

Fig. 2. Phantom conjunctions in a 24-h window for 21,000 TLE objects. An altitude masque of 500–600 km was used. Old TLE data of 100 Starlink satellites played the role of phantoms. Th: Threshold. Each phantom was associated with waypoints of a 1-Hz sampling rate, i.e. 86,400 (= 3600×24) points. (a) Computation time. Threshold difference has a miniscule effect. Top black: Current code. Middle blue: The STELA DB on memory. Red: Expected after re-engineering. (b) Number of conjunctions of a phantom with respect to threshold distances.

want to determine its miss distance against the entire catalogue of object. Let δ_{ijk} be the distance between x_{ij}, i.e., the j-th waypoint of trj_i and the k-th object of the catalogue at time t_j. Let y_i be a binary decision variable, such that $y_i = 1$ if trj_i is selected; otherwise, $y_i = 0$. In addition, we add a constraint of maximum propellant consumption Ψ. This formulation can then be expressed as a constrained linear integer programming problem:

$$\underset{i \in TRJ}{\mathrm{Max}} \sum_{j \in trj_i} \min_{k \in CAT} \left\{\delta_{ijk} \cdot y_i\right\} \tag{3}$$

$$y_i \cdot \sum_{j \in trj_i} \left\{\psi_{ij}\right\} \le \Psi \;\; for\ i\ = 1, 2, \cdots, n. \tag{4}$$

where y_i is a binary decision variable for $i = 1, 2, \cdots$, n and CAT is the catalogue set. The subscripts and set symbols are intentionally abused for notational simplicity. Consider a trajectory trj ∊ TRJ. To determine the miss distance of trj, the DCA of each of the m waypoints of trj must be calculated. Calculation of the DCA of a waypoint wp requires CA of wp against all catalogue objects when wp is defined. Hence, for a 1-Hz sampling rate over 24 h with 21,000 catalogue objects, each trajectory requires more than 1.8 billion (specifically 1814,400,000 = 86,400 * 21,000) distance calculations between two points unless an efficient method is employed. This cost is linear with respect to catalogue size.

However, the Voronoi diagram helps significantly reduce this computation. For a phantom following trj, the miss distance, i.e., the DCA, is defined with one of the catalogue objects in the Voronoi neighbourhood over the timeline. Given a 3D Voronoi diagram, the average number of neighbouring objects sharing a Voronoi face is usually 15–16 for a random object distribution. The 2D and 3D kissing numbers are 6 and 12, respectively. Suppose that 20 is the relaxed upper bound. Then, less than 1.8 million (specifically 1728,000 = 86,400 * 20) distance calculations are required for each trajectory. The Voronoi diagram thus yields more than a 1000-fold computation reduction for each trajectory. Notably, the cost of the Voronoi-based approach is independent of catalogue size.

Formulation 2. Optimization of the weighted sum of miss distance and propellant consumption: Unconstrained integer linear programming problem

If the propellant constraint is moved into the objective function, we have a multi-objective optimization problem that aims to maximize the minimum distance and minimize propellant usage. In general, a multi-objective optimization problem results in a set of points in the objective function space, referred to as the Pareto optimal solutions, which lie on the Pareto front (the set of all non-dominated solutions). In this case, the speed of calculation for δ_{ijk} would facilitate the solution process.

$$\underset{i \in TRJ}{\mathrm{Max}} \left\{ \sum_{j \in trj_i} w_1 \cdot \min_{k \in CAT} \left\{\delta_{ijk} \cdot y_i\right\} + \; w_2 \cdot \left\{ \sum_{j \in trj_i} \psi_{ij} \right\} \cdot y_i \right\} \tag{5}$$

6. K-nearest neighbors in timeline

Consider a phantom following trajectory trj. We want to find the catalogue object that approaches closest to the phantom during the window. The object, o_{1t}, can be found by checking the distance between its waypoint and the locations of the catalogue objects, both at t in the window. Let us denote o_{1t} as 1-nn denoting the closest neighbour. The red curve in Fig. 3(a) shows the profile of the distances between o_{1t} and its nearest neighbour, 1-nn, every second. The red star denotes the minimum and defines the miss distance of the phantom. Fig. 3(b) and (c) show the distance profiles of the phantoms, where the miss distances are the middle and largest amongst the miss distances defined by all phantoms. Stars denote the minima of the curves. Distance profiles were produced using the CA algorithm.

Given a phantom, suppose that we want to find the object o_k that defines the kth smallest distance to the phantom over the window, called k-NN. The solution process for the phantom conjunction problem produces a k-NN as a by-product. Fig. 3(d) shows the closest approach distance (CAD-1) graph of the 100 phantoms. The horizontal axis denotes the different phantoms. The vertical axis denotes the missing distance for each phantom. Fig. 3(e) shows the CAD-k graph, where $k = 3$ and the blue and green curves correspond to the second and third closest approaching objects to each phantom, respectively. Fig. 3(f) shows the weighted sums of the curves. CAD-w3 corresponds to the uniformly weighted sum of the three CAD-3 curves. CAD-w5 similarly corresponds to CAD-5. All these curves can be produced with negligible additional computation.

The optimal COLA formulations are deterministic. An optimization model can be defined to incorporate uncertainties using the nearest-neighbour information available in the k-NN output.

SpaceMap technology is based on the STELA DB. Hence, the STELA has to be constructed efficiently, concisely, and robustly. This goal applies to both offline and online constructions. “Offline” indicates that the batch construction of STELA occurs on regular basis, e.g., three times a day with the entire catalogue downloaded from the SSA databases, such as the Space Track TLE database. “Online” indicates the incremental update of STELA by reflecting marginal changes in the catalogue state, where the change implies (A) state changes of some catalogue objects, (B) deletion of some catalogue objects, and (C) insertion of new objects to the catalogue. Several use cases exist for real-time online database updates. A variation of the phantom conjunction algorithm can be used for this purpose.

7. SpaceMap’s 42 services for space industries

7.1. Vision

SpaceMap’s vision is to create a safer, more efficient, and sustainable space environment. Our mission is to realize this vision by developing and promoting real-time algorithms for many, if not all, critical decision-making problems for space assets using moderate computational resources. SpaceMap provides three access points for users. AstroLibrary is intended for application program developers. Astro-1 and AstroOrca are the other two resources for owners and operators on the platform. Astro-1 is a real-time unified CA/COLA solution, and AstroOrca is used for complex space optimization and intelligence problems (Fig. 4).

SpaceMap offers two orthogonal perspectives to space sustainability: physical and financial. For space to become a sustainable and valuable natural resource for humanity, collisions should be predicted and avoided to the extent possible. We address this challenge with Astro-1. For the space economy to be sustainable, investments should be sustainable. Prior investments should be profitable to ensure sustainable investment. For an investment to be profitable, the utilization of the invested resources must be optimized. Owing to the extreme conditions of the space industry (i.e., extreme speeds for both object flights and catalogue size increases), optimization problems are difficult to solve. However, there are many space optimization problems. Methods for solving these problems require rapid and easy development. A good library is necessary for sustainable investment in space. AstroLibrary provides the answer to this question. A historical parallel is the development of math libraries (e.g., math.h) in computer programming in the 1980s. It is not necessary to “reinvent the wheel” in space.

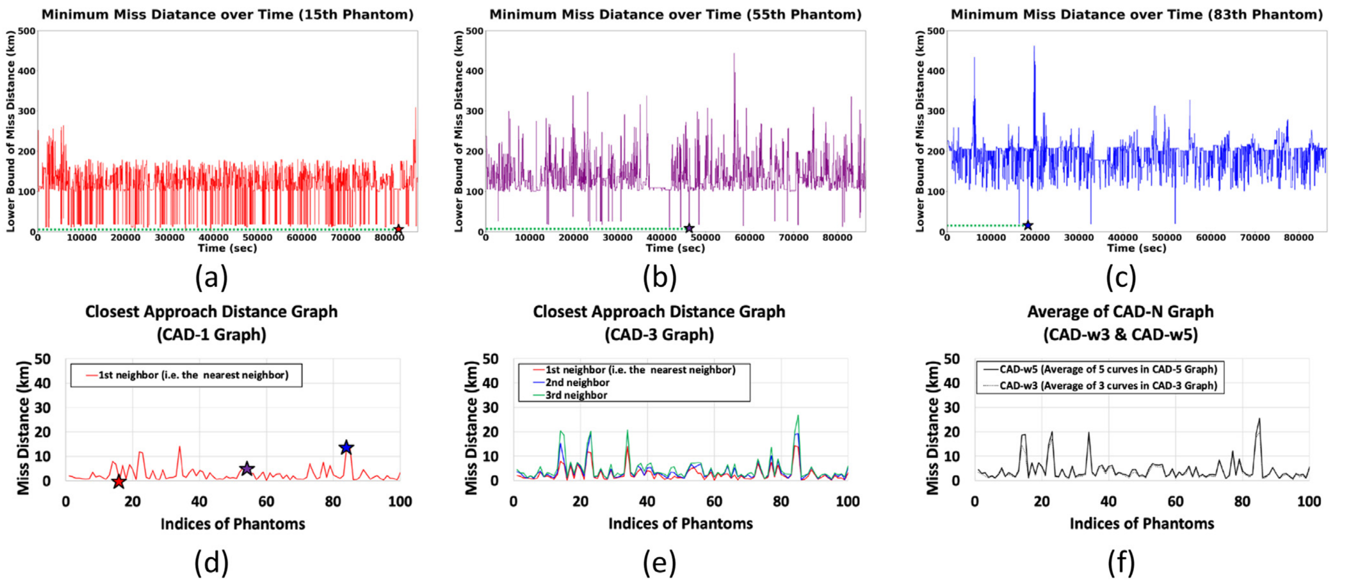

Fig. 3. Nearest neighbors and their applications. (a)–(c) Nearest neighbour profiles of three phantoms over the window. Horizontal axis: Timeline (24 h). Vertical axis: Lower bound of miss distance to the nearest neighbour. Different colors correspond to the different phantoms. Red: The phantom with the smallest miss distance of all phantoms over the time. Purple: The phantom with the 70th smallest miss distance of all phantoms over the time. Blue: The phantom with the largest miss distance of all phantoms over the time. (d)–(f) Closest approach distance graphs of the 100 phantoms. Horizonal axis: different phantoms. Vertical axis: Miss distance of each phantom against the entire catalogue of objects during the window. (d) Closest approach distance graph. (e) CAD-3 graph, i.e., the distances to the third-closest approaching objects. (f) Weighted sum of the k curves in CAD-k.

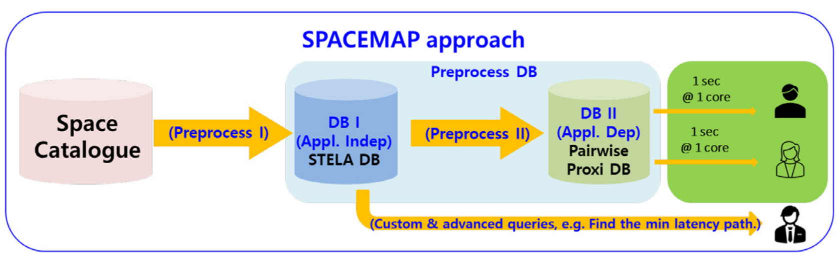

Fig. 4. SpaceMap solution to the following fundamental question: Where should we allocate more computational resources and where should we achieve real-time responses?.

Table 1
Three service types of AstroLibrary and their features.

	Library Type (A)	Programming Skill Needed (B)	Computing Platform (C)	Data Security (D)	Price (E)	Maintenance (F)
1	RESTful API	Low	SpaceMap	W/ Network	Low	Easy
2	Python Library	Low	SpaceMap Customer	W/ Network W/O Network	Medium	Medium
3	C++ Library	High	Customer	W/O Network	High	Demanding

7.2. Using AstroLibrary

SpaceMap functions are available as APIs in AstroLibrary. Anyone with moderate programming skills can develop application programs using these APIs. The spatiotemporal Voronoi engine is written in C++. There are three types of libraries in AstroLibrary (Table 1). The Python and C++ versions can be installed in the programmer's computing resources and securely disconnected from the Internet, if necessary. The RESTful web API works as follows: When the API embedded in the user's application program is executed, SpaceMap's computing server produces solutions. Table 1 summarizes the three library types in AstroLibrary and compares their features. AstroLibrary can be downloaded from GitHub (https://github.com/SPACEMAP42/AstroLibrary). The Python package is set to be released on the Package Installer for Python (PIP).

8. Conclusions

In this paper, we presented the real-time CA/COLA features of SpaceMap, which is designed for O(10^6) or more objects. The theoretical and computational basis is the Voronoi diagram, which is the most concise and efficient data structure for spatiotemporal reasoning amongst many objects in 2D and 3D space. The produced STELA DB is the core of the ST AI. The algorithms implemented in C++ are available on GitHub as AstroLibrary, which has RESTful APIs and a Python package. Using this library, efficient application programs can be developed easily and conveniently for challenging spatiotemporal problems by anyone with elementary programming skills. We expect the proposed algorithm to replace the seminal three-filter algorithm, which is computationally expensive.

The three-filter algorithm is intuitive and easy to implement. Graduate students with moderate programming skills can implement the program within a few months, if not weeks, and produce reasonably good conjunction reports. Owing to this characteristic, companies tend to implement the three-filter algorithm independently with slight differences in acceleration methods, programming language environments, and other aspects. We have observed this phenomenon in many companies, including in a recent report [25]. However, significant resources are consumed to make its implementation efficient and robust. Regarding CA/COLA, "reinventions of the wheel" seem common. Its consequence is noteworthy: it is costly and time-consuming, and the solution quality is questionable.

Declaration of competing interest

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

CRediT authorship contribution statement

Shawn SH Choi: Data curation, Formal analysis, Software, Writing – review & editing. **Peter JH Ryu:** Formal analysis, Writing – review & editing. **Kyuil Sim:** Software. **Jaedong Seong:** Validation. **Jae Wook Song:** Formal analysis, Methodology, Validation. **Misoon Mah:** Validation. **Douglas DS Kim:** Conceptualization, Project administration, Supervision, Writing – original draft.

Acknowledgments

This research was supported by the National Research Foundation of Korea (NRF, Grant No. 2017R1A3B1023591), the Korea Institute of Startup and Entrepreneurship Development (KISED, Grant No. RS-2023-00283180), and the Institute for Information and Communication Technology Planning and Evaluation (IITP, Grant No. RS-2023-00259061). This research was partially supported by the Asian Office of Aerospace Research and Development (AOARD, Grant No. FA2386-22-1-4067).

References

[1] A. Falle, et al., One million (paper) satellites, Science (1979) 382 (2023) 150–152.
[2] ESASpace Environment Statistics, 2023 https://sdup.esoc.esa.int/discosweb/statistics/ accessed 29 Oct.
[3] Space Situational Awareness (SSA) and Space Traffic Management (STM) Workshop, 2023 https://www.youtube.com/watch?v=gW4aW0gkWc4 accessed 31 Oct.
[4] Donald J. Kessler, G. Burton, Cour-Palais, Collision frequency of artificial satellites: the creation of a debris belt, J. Geophys. Res. 83 (A6) (1978) 2637–2646.
[5] Space-trackSpace-Track, 2023 https://www.space-track.org/ accessed 14 June.
[6] F.R. Hoots, et al., An analytic method to determine future close approaches between satellites, Celest. Mech. 33 (1984) 143–158.
[7] J. Alarcón Rodríguez, et al., Collision risk assessment with a smart sieve method, in: Joint ESA-NASA Space-Flight Safety Conference, 486, 2002, p. 159.
[8] I. Budianto-Ho, et al., Scalable conjunction processing using spatiotemporally indexed ephemeris data, Advanced Maui Optical and Space Surveillance Technologies Conference, 2014.
[9] B. Abernathy, et al., The CAOS-D architecture for conjunction analysis, Infotech@ Aerospace 2011 (2011) 1434.
[10] IARPASpace Debris Identification and Tracking (SINTRA), 2023 https://www.iarpa.gov/research-programs/sintra accessed 29 Oct.
[11] NASANASA Seeks Solutions to Detect, Track, Clean Up Small Space Debris, 2023 https://www.nasa.gov/coeci/ accessed Oct 29.
[12] S. Nag, et al., Prototyping operational autonomy for space traffic management, Acta Astronaut. 180 (2021) 489–506.
[13] J. Cha, et al., DVD-COOP: innovative conjunction prediction using Voronoi-filter based on the dynamic Voronoi diagram of 3D spheres, in: Advanced Maui Optical and Space Surveillance Technologies Conference, 2017, p. 98.
[14] S.S. Choi, et al., SPACEMAP: real-time web server for safer, more sustainable and efficient space, in: Advanced Maui Optical and Space Surveillance Technologies Conference, 2022, pp. 27–30.
[15] S.S. Choi, et al., Conjunction assessments of the satellites transported by KSLV-II and preparation of the countermeasure for possible events in timeline, J. Space Technol. Appl. 3 (2023) 118–143 in Korean.
[16] C. Song, et al., Dynamic Voronoi Diagram for Moving Disks, IEEe Trans. Vis. Comput. Graph. 27 (2021) 2923–2940.
[17] Tim Flohrer, et al., Assessment and categorization of TLE orbit errors for the US SSN catalogue, Risk. 8 (2008) 10–11.
[18] S. Singh, A new tool for conjunction analysis of ISRO's operational satellites, close approach prediction software: CLAPS, J. Space Saf. Eng. 7 (2020) 290–294.
[19] David A. Vallado, Paul J. Cefola, Two-line element sets–practice and use, 63rd International Astronautical Congress, 2012.
[20] Rowan Philip Jansens, Improving SGP4 Orbit Determination with New State Estimation Algorithm, Los Alamos National Lab.(LANL), Los Alamos, NMUnited States, 2022 No. LA-UR-22-28307.
[21] G. Curzi, et al., Two-line-element propagation improvement and uncertainty estimation using recurrent neural networks, CEAS. Space J. 14 (1) (2022) 197–204.
[22] CelestrakSOCRATES, 2023 https://celestrak.org/SOCRATES/ accessed 30 Oct.
[23] S. Alfano, Relating position uncertainty to maximum conjunction probability, J. Astronaut. Sci. 53 (2005) 193–205.
[24] L. Chen, et al., Orbital Data Applications for Space Objects, Springer, 2017.
[25] , Small Satellite Conference 2022, 2022 Available at https://smallsat.org/ accessed.