DVD-COOP: Innovative Conjunction Prediction using Voronoi-filter based on the Dynamic Voronoi Diagram of 3D Spheres

Jehyun Cha[1], Joonghyun Ryu[2], Mokwon Lee[1], Chanyoung Song[1], Youngsong Cho[2], Paul Schumacher[3], Misoon Mah[3], Deok-Soo Kim[1,2,*]

[1] School of Mechanical Engineering, Hanyang University, Seoul, Korea

[2] Molecular Geometry and Voronoi Diagram Research Center, Hanyang University, Seoul, Korea

[3] US Air Force Research Laboratory (AFRL), USA

** Corresponding author(dskim@hanyang.ac.kr)*

ABSTRACT

Conjunction prediction is one of the critical operations in space situational awareness (SSA). For geospace objects, common algorithms for conjunction prediction are usually based on all-pairwise check, spatial hash, or kd-tree. Computational load is usually reduced through some filters. However, there exists a good chance of missing potential collisions between space objects. We present a novel algorithm which both guarantees no missing conjunction and is efficient to answer to a variety of spatial queries including pairwise conjunction prediction. The algorithm takes only $O(k \log N)$ time for N objects in the worst case to answer conjunctions where k is a constant which is linear to prediction time length. The proposed algorithm, named DVD-COOP (Dynamic Voronoi Diagram-based Conjunctive Orbital Object Predictor), is based on the dynamic Voronoi diagram of moving spherical balls in 3D space. The algorithm has a preprocessing which consists of two steps: The construction of an initial Voronoi diagram (taking $O(N)$ time on average) and the construction of a priority queue for the events of topology changes in the Voronoi diagram (taking $O(N \log N)$ time in the worst case). The scalability of the proposed algorithm is also discussed. We hope that the proposed Voronoi-approach will change the computational paradigm in spatial reasoning among space objects.

1. INTRODUCTION

Many objects orbit about the Earth. E.g. artificial satellites, natural satellites, space debris, etc. As of 2013, more than 6,600 satellites have been launched since Sputnik 1 in 1957: 3,600 remained in orbit [1] including about 1,000 operational ones [2]. Space debris are man-made objects in the Earth orbit such as defunct satellites, rocket stages, fragments from disintegration and collisions, etc. According to European Space Agency (ESA), there were approximately 29,000 large debris of size bigger than 10 cm, 670,000 debris of size 1~10 cm, more than 170 million ones smaller than 1 cm (as of July 2013) [3].

Orbital objects move at high speed of up to 16 km/s if head-on: E.g. the 2009 satellite collision occurred at 11.7 km/s. As the impact of collision between space objects can be catastrophic, it is necessary to monitor the trajectory of space objects. The detection, tracking, identification, cataloging, etc., altogether frequently referred to as space situational awareness (SSA), of all observable objects in the Earth orbit is currently done by the United States Strategic Command's (USSTRATCOM) Joint Space Operations Center (JSpOC, https://www.space-track.org/) [4]. The ephemeris of the objects is stored in the TLE (Two-Line Element) format in Space Catalog of JSpOC which catalogued 16,636 objects as of August 8, 2017: This number is projected soon to be more than 100,000. The rapid increase of the number of cataloged objects is due to both the increase of object population and the advancement of sensor technology to detect smaller ones.

SSA is becoming more important as the number of space objects increases. The collision between objects is an immediate consequence of the high density of objects and it accelerates density increase. E.g. the accidental satellite collisions (e.g. Iridium 33 vs. Kosmos 2251 in 2009; Produced more than 2,000 catalogued debris), the planned anti-satellite missile test (e.g. Chinese Fungyun satellite destroyed by a missile in 2007; Produced more than 2,000 trackable ones among 150,000 debris), new satellite launches, etc. Collision risks in space increases despite of the effort of post-mission disposal of vehicles [5]. Securing a perfect SSA is both costly and complicated.

One of the critical SSA issues is the prediction of conjunction which is a dangerous condition of co-located objects within a near proximal distance, referred to as conjunctions, in which case the probability of collision rapidly increases.

It is necessary to monitor and predict accurately if any two objects approach within a close proximity so that a decision can be correctly made for evasive maneuver of either or both. There were abundant studies on conjunction prediction: Studies for measurement, estimating orbits, predicting inter-object distances, etc. One of the key research issues is the reduction of the search space in the $O(N^2)$ combinations of N objects to check inter-object distances at an arbitrary moment. This effort to reduce search space is critical in that i) object population increases very rapidly and ii) diverse spatial reasoning queries including conjunction prediction will have to be performed much more frequently than it is now.

In this paper we propose an algorithm, with an implementation, to predict conjunctions among the objects in JSpOC Space Catalogue. The proposed DVD-COOP (Dynamic Voronoi Diagram-based Conjunctive Orbital Objects Predictor) algorithm/program is event-based, general purpose (beyond pairwise conjunction prediction), efficient, accurate, and coordinate system independent. The algorithm is based on the Voronoi diagram of moving three-dimensional spheres. Its computation result is efficiently re-playable for diverse analyses on the fly. Summary - The properties of the proposed algorithm:

- General purpose
- Independence of coordinate system
- Efficient replay
- Event-based
- Analytic and efficient

The Voronoi diagram used in this paper is a generalization of the ordinary Voronoi diagram of points in that the generators are three-dimensional spheres, not points. The Voronoi diagram in this paper is dynamic in that the spherical generators move through respective orbits, some generators may disappear, may be created, or a generator may split to multiple generators, etc. Computational statistics is presented using the ephemeris data of the JSpOC Space Catalogue. Experiment was performed using a single core. This paper presents the conjunction problem from nominal geometry point of view and uncertainties related with covariance [6] are left for future works. The rationale behind the parallelizability of the proposed algorithm is also presented. We expect the proposed Voronoi diagram approach not only improves computational efficiency for predicting conjunctions but also efficiently finds answers to a variety of spatial queries. Time complexity in this paper is all for the worst case unless otherwise stated. We hope the Voronoi diagram approach will contribute to transform the computational paradigm of SSA.

2. LITERATURE SURVEY

Conjunction has two interpretations in literature [7]. Point of closest approach (PCA) refers to a local minimum of the distance between two objects such that the distance is less than an a priori defined threshold. Conjunction interval (CI) refers to the time interval during which the inter-object distance objects is less than the threshold. The PCA interpretation is common in literature and software. This paper discusses conjunction in the PCA concept.

Accurate and efficient computation for conjunction prediction is critical. The value of conjunction prediction is obvious in that, if a conjunction is correctly predicted, a collision-avoiding evasive manoeuvre of an object can be planned and executed. When it is possible, an optimal manoeuvre pathway might need to be determined or designed by possibly evaluating the effect of each candidate manoeuvre to future conjunctions. The formulation of this optimization problem involves parameters that can be obtained from multiple executions of conjunction prediction, each time with modified ephemeris. This implies a very high frequency of executing conjunction prediction than it is done today. E.g. Collision Risk Assessment tool (CRASS), developed by GMV/ESA, forecasts conjunctions on a daily basis with a prediction time window of one week, a policy made in the consideration of a trade-off between orbit prediction accuracy and reaction time to a predicted conjunction [8, 9].

A naïve approach to conjunction prediction might be as follows. Suppose that Space Catalogue has N objects (or equivalently N orbits) and thus $O(N^2)$ object pairs. It is easy to make linear approximations of two orbits corresponding to some discrete moments in time. Consider a line segment L with an approximation error ε. Let $\hat{O}$ be the replica of an object O, for notational convenience, with a null size: $\hat{O}$ moves through L while O moves through its elliptic orbit. Let S be a sphere, called a **replica sphere**, with the radius ε. It can be easily shown that, at an arbitrary moment of time, if two objects are conjunctive, the replica spheres corresponding to the two objects are also conjunctive. Hence, a real conjunction can be detected by checking the intersection between two replica spheres enlarged to appropriate

sizes according to the linear approximation errors of paths. Without loss of generality, it can be assumed that each full elliptic orbit is approximated by M_L line segments. This leads to $O(M_L N^2)$ pairs of line segments to be tested for the interference between two enlarged moving replica spheres.

How to check the intersection between two replica spheres under linear motions in the naïve approach? Suppose that some point locations are sampled from the two line segments, say L_1 and L_2, corresponding to replica spheres S_1 and S_2, respectively, and the replica spheres are placed at these sampled locations to check if they intersect each other or not. Hence, it takes $O(M_R)$ intersection tests for a sphere pair for M_R sampling points on each line segment (assuming that moving speed is properly considered). Hence, it takes $O(M_L M_R N^2)$ time for a single complete elliptic orbit for each of all the space objects. Considering the size of satellite compared to the lengths of L_1 and L_2 (i.e. the size vs. traveling distance of satellites), M_R should be very big. The main idea of this naïve method can be said to enumerate the space as effectively as possible with sampled points in the absence of efficiency consideration.

The naïve approach is to check all pairwise objects for all possible moments – This is infeasible due to computational requirement. A trade-off exists between sampling frequency and computation cost: A high computation cost is required in order to reduce the chance of missing conjunction (as shown in Fig. 1(b)). Be aware that this approach cannot eliminate the possibility of missing intersection regardless how high a sampling frequency is.

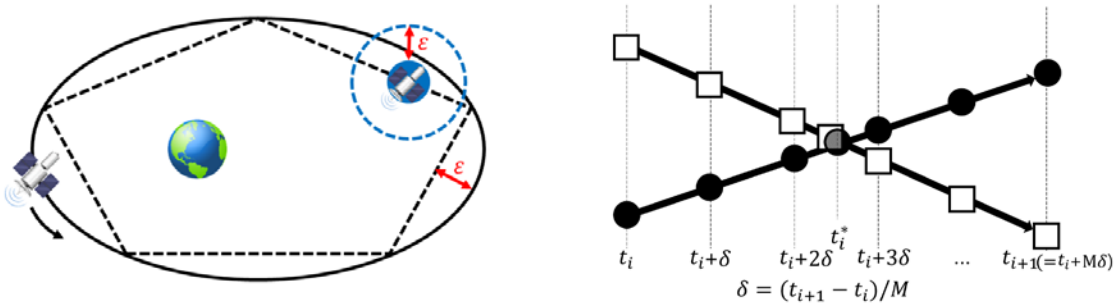


Fig. 1. Naïve approach for conjunction prediction. (a) Piecewise linear approximation of an orbit with five segments. (b) Enumerating locations on the linear approximation with a finite points seperated by δ in time. The collision at t_i^* is not detected by the suggested sampling.

To reduce the computational requirement of the naïve approach, many studies were conducted. Most existing studies were about the reduction of search space by filtering out orbit combinations that were guaranteed to be free of any conjunction [10, 11, 12, 13, 14, 15, 16, 17, 18, 19] which were the improvements of the initial idea of the three-filter approach proposed by Hoots et al. (1984) [20]: Apogee/perigee filter (Filter I), orbit path filter (Filter II), and time filter (Filter III). Note that the first two filters (I and II) are time-independent and geometric. Casanova et al. (2014) reported the following [10]: Given 372,816 (=864*863/2) object pairs from 864 objects, Filter I reduces the number of pairs down to 218,459 and Filter II further reduces to 49,739 pairs, resulting into a total of 86.7% reduction of object pairs in the explicit intersection test. Filter III needs to be applied to the rest, 13.3%.

A natural effort to speed-up the computation efficiency would be to use a spatial hash which was devised for a more direct access to objects thus hopefully reducing the search space of the naïve approach [4]. The spatial hash algorithm in [4] suggested to define a three-dimensional grid system, say G, which should include all orbits and contain a sufficient number of buckets which were represented as a three-dimensional array. Let δ be the time step of each prediction time slice. During δ, each object orbits through an elliptic arc, say ξ, whose positions and tangent vectors corresponding to the beginning and ending of δ provide a planar triangle T on the plane of travel in the three-dimensional space. We observe that this triangle is identical to the control polygon of the rational quadratic Bezier curve of the elliptic arc. Hence, for each orbit arc (and thus each object), the elements of the grid system G which intersect the triangle T can be marked. Hence, the grid element marked by both objects only needs to be further studied for the possible conjunction. [4] reported an experimental result: An object in Low Earth Orbit (LEO) with a 60 s time step required a bounding box with dimensions of over 400 km whereas a grid cell dimension was 50 km by 100 km. Spatial hash requires to fine tune several parameters.

Relatively recently, kd-trees were introduced for conjunction prediction. Kd-trees recursively subdivide in principle both the set of data points and the corresponding Euclidean space into progressively finer subsets and sub-regions. A node of the tree corresponds to both a subset of the space and a subset of the input data points. Given a height-balanced kd-tree constructed in $O(N \log N)$ time and stored in $O(N)$ memory, an insertion or a nearest neighbor search could be done in $O(\log N)$ time on average. Many other queries can also be done very efficiently.

The idea of using a kd-tree in conjunction prediction is to extend that of spatial hash in that the trajectory of each orbit during a time step δ in the prediction time window is a basic unit of information. For each of *X*-, *Y*-, and *Z*-axis motions, both the minimum and maximum of travel interval is stored in a six-dimensional kd-tree. Two objects whose trajectory during δ do not overlap is safely filtered out from further study of conjunction. As far as we are aware of, the initial use of a kd-tree in conjunction prediction was the Moving Object Processing System (MOPS) system of University of Hawaii's Pan-STARRS began in 2002 [21, 22, 23, 24]. Kd-trees were used to efficiently answer a variety of spatial queries in conjunction prediction [25]. CAOS-D system also employed the kd-tree for conjunction prediction [4].

Summary: The previous approaches have the following properties:

- Tailored for pairwise conjunctions
- Coordinate system dependency
- Difficulty to replay
- Time increment-based

The previous approaches seem not easy to replay computation results in order to answer other spatial queries than that was done during the computation. One of the reasons is each kd-tree (or spatial hash) corresponding to each time step δ requires O(N) memory. It seems that the kd-tree and spatial hash methods can find all conjunctions if correctly implemented while the naïve method may miss a conjunction.

It is also necessary to review computer programs in conjunction prediction. CRASS of GMV/ESA employed the Smart Sieve method which was based on various filters introduced by Hoots et al. in 1984 (and Klinkrad in 1993) and their variations [26]. CAOS-D (Continuous Anomalous Orbital Situation Discriminator) developed by Intelligent Software Solutions for AFRL used both the Conjunction Sieve tool developed by Aerospace Corporation and the Smart Sieve method [4]. A report on the comprehensive experimental benchmark among five programs, including CAOS-D and CSieve, for conjunction prediction is available [7].

We emphasize that there must be other types of queries, beyond pairwise conjunction, in diverse applications which are both theoretically more involved and computationally more demanding: E.g. triplet-wise and quadruplet-wise conjunctions. Another useful capability might be a quick and efficient replay of a prediction result with different analyses.

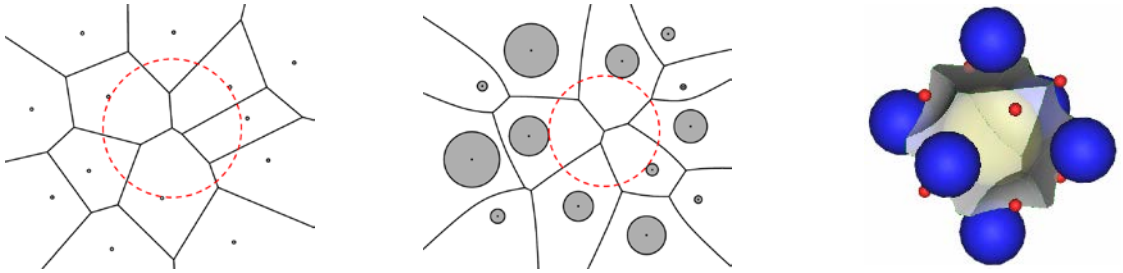
3. VORONOI FILTER

3.1. What is a Voronoi diagram?

Given a set $P = \{p_1, p_2, ..., p_N\}$ where p_i is a point generator in a *d*-dimensional space, the **ordinary Voronoi diagram** $Vor(P)$ of P is a tessellation of the space where each cell of the tessellation is the set of locations where each location is closer to the generator of the cell than to the other generators [26, 27]. Usually the Euclidean distance is employed. Fig. 2(a) shows an example of such an ordinary Voronoi diagram in the plane. The boundary between two adjacent point generators is the perpendicular bisector of the two generators. Given a Voronoi diagram, many spatial queries can be efficiently and correctly (or at least most accurately) answered. The red dotted circle is the biggest empty circle that can be placed in the middle of the point generators where the empty circle does not contain any input generator. This query can be correctly answered in $O(N)$ time if $Vor(P)$ is available. The ordinary Voronoi diagram has long been used for efficiently solving spatial reasoning problems in a variety of disciplines. $Vor(P)$ can be constructed in $O(N^2 \log N)$ time for $d = 3$ ($O(N \log N)$ for $d = 2$) in the worst case. Note that the worst case combinatorial complexity of $Vor(P)$ is $O(N^2)$. The average time complexity of $Vor(P)$ is $O(N)$ for *d* both 2 and 3. However, its robust construction is in practice of critical concern. $Vor(P)$ can be robustly constructed by the topology-oriented incremental (TOI) algorithm in $O(N^3)$ time in the worst case but $O(N)$ time on average.

Let $B=\{b_1, b_2, \ldots, b_N\}$ where b_i is a spherical ball in the d-dimensional space where $b_i \cap b_j = \phi$, $i \neq j$. The **Voronoi diagram** $VD(B)$ of B can be similarly defined as a tessellation of the space where each cell of the tessellation is the set of the locations where each location is closer to the boundary of the corresponding ball than to the other balls. In VD where $d = 3$, we refer the set of locations equidistant to a pair of balls as a Voronoi face, abbreviated as a V-face, that to three balls a V-edge, and that to four balls a V-vertex. It is relatively recent that important properties of this type of Voronoi diagram for $d = 2$ and 3 were discovered and the construction algorithms were devised [28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38]. Fig. 2(b) shows an example in the plane with the biggest empty circle found in O(N) time. There are many important applications that can be efficiently solved using this type of Voronoi diagram. Fig. 2(c) shows the Voronoi diagram of spheres where $d = 3$.

Numerical robustness is one of the most fundamental challenge of the Voronoi diagram of circles and spheres yet to be studied. No robust Voronoi diagram construction algorithm has been known for 3-dimensional spheres while its counterpart in the plane was recently reported with an implementation [38]. We have recently succeeded to extend our TOI-algorithm of the plane to the three-dimensional space.



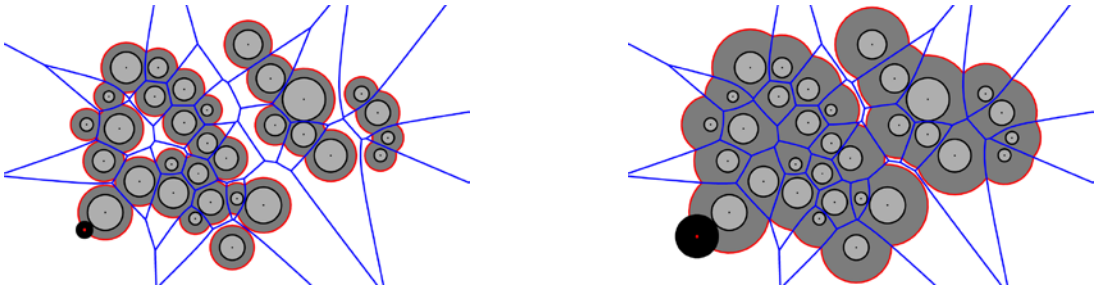
(a) (b) (c)

Fig. 2. Voronoi diagrams (a) Voronoi diagram of points (2D). (b) Voronoi diagram of circular disks (2D). (c) Voronoi diagram of spherical balls (3D)

3.2. How useful are Voronoi diagrams?

Fig. 3 shows an application of offset of the Voronoi diagram of disks in the plane to demonstrate its capability: It shows how the Voronoi diagram can be used for solving difficult application problems correctly, efficiently, and conveniently. Suppose that a set of N input disks is given (Fig. 3, the light grey circles). The locus of the center of a circular black probe touching the input disks defines the offset of constant distance (the red piecewise circular arc curve in Fig. 3(a)). Its computation takes $O(N^2)$ time if we do a naïve check between the offset circles of each input disk pair and the algorithm is complicated to get correct solution. However, the intersection between two offset arcs occurs on the Voronoi edge of the Voronoi diagram of the input disks. Hence, if the Voronoi diagram is available, the correct offset can be computed in $O(N)$ time simply by checking each V-edge in the Voronoi diagram. The algorithm is simple and fast. In addition, the existence of void structure within the left disk cluster and the tunnel structure between two clusters can also be easily recognized. Fig. 3(b) shows that the same Voronoi diagram can be used for a bigger probe: The fact that neither void nor tunnel exists can be also recognized. Similar observation holds for its three-dimensional counterpart for the arrangement of spherical atoms and various application problems were solved, particularly for molecular problems [39, 40, 41, 42, 43, 44, 45, 46]. The following summary is well-known.

- Voronoi diagrams are useful for accurately, efficiently, and conveniently solving spatial reasoning problems among particles.



(a) (b)

Fig. 3. $O(N)$ time computation of offset curves with the Voronoi diagram. An offset curve corresponds to the center of circular probe (black). (a) Small probe. A void and two tunnels are recognized. (b) Bigger probe. Both void and tunnels disappeared.

3.3. Dynamic Voronoi Diagram

Would the useful properties of Voronoi diagrams extend to the case where some particles move? If it does, how could we achieve it? In summary, the useful properties extend to the dynamic case if the **dynamic Voronoi diagram** can be efficiently constructed.

(An example: 2D) Fig. 4 shows the Voronoi diagrams of disk generators where one disk moves to the right according to its velocity vector indicated by the black arrow. Each of the eight Voronoi diagram instances is associated with its time stamp t_0, t_1,…, t_7. The algorithm of the dynamic Voronoi diagram (DVD) algorithm detects the moments where critical events occur to maintain a correct Voronoi diagram. The DVD algorithm begins with the initial Voronoi diagram VD_0 at t_0. The topology of VD_1 at t_1 is identical to that of VD_0 whereas the topology of VD_3 is different from that of VD_0. So, we compute the moment t_2 when the topology changes. Note that this moment corresponds to the moment that a V-edge of VD_0 shrinks to a point and at the same time the shrunk point begins to grow to another V-edge at t_3. Hence, t_2 is the moment of edge-flipping in that a V-edge of VD_0 flips in the topology structure. In the example, there are two more edge-flipping moments as indicated by the big blue ellipses. It is known that the edge-flipping moment can be computed by solving an 8-th degree polynomial for four disks of arbitrary sizes which linearly move with arbitrary speed in the plane. In three-dimensional space, the flipping moment from a V-edge to a V-face or vice versa can be computed by solving a 10-th degree polynomial [47, 48]. Correctly solving the root-find problem of these polynomials is not trivial at all.

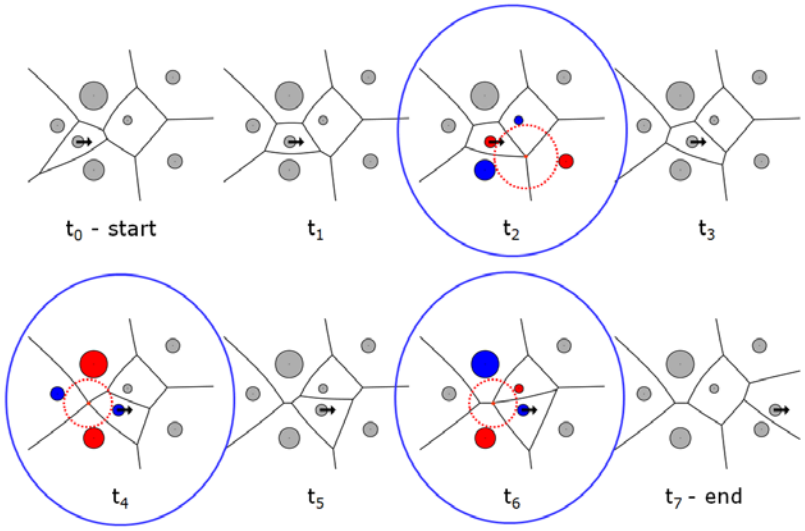


Fig. 4. Dynamic Voronoi diagram where one disk linearly moves. (t_0) Initial state. (t_1) before the first V-edge flips, (t_2) the moment when a V-edge flips (i.e. a V-edge shrinks to a point). (t_3) after the V-edge flips. (t_4) another V-edge flips. Similar observations follow in (t_5) (t_6) and (t_7).

(DVD-algorithm: 3D) Let N be the number of spheres in the three-dimensional space. The DVD algorithm of spheres consists of three major steps:

i) Construct the initial Voronoi diagram.
ii) Compute the initial events of the Voronoi diagram.
iii) Process the most immediate event in future, propagate its influence in the neighbor, and update the event set.

The algorithm repeats Step iii) for a sufficient number of times. Step i) takes $O(N^3)$ time in the worst case but $O(N)$ on average, particularly for the conjunction prediction problem. Be aware that the combinatorial complexity of the Voronoi diagram in the three-dimensional space is already quadratic: I.e. $M = O(N^2)$ where M represents the number of V-vertices, V-edges, and/or V-faces. However, $M = O(N)$ on average. It is safe to consider the average case in conjunction prediction. Step ii) takes $O(M \log M)$ time if the events are stored in a priority queue implemented by heap data structure where each of its nodes corresponds to a V-edge using the event time as a key. Step iii) takes $O(K \log M)$ time for processing K events during the prediction time window. This is because the processing of each event from the queue takes $O(\log M)$ time if the heap is correctly maintained (i.e. height-balanced). Note that $K \propto N$, $K \propto V$, and $K \propto T$ (V: the velocity of objects; T: the length of the prediction time). Both the Voronoi diagram and priority queue require $O(M)$ memory. The events occurred over the prediction time horizon is stored in an event history queue which takes $O(K)$ memory. We store the predicted history of consecutive flipping events in the **COOP-HSTRY file** taking $O(K)$ memory.

Given the COOP-HSTRY file, the topology structure of the Voronoi diagram of any arbitrary moment t^* in the prediction window can be efficiently produced (i.e. $O(k)$ time for k flipping events before t^*) by scanning the appropriate portion of the flipping history. Then, the construction of the Voronoi diagram at t^* requires only $O(N)$ time with the valid topology structure information.

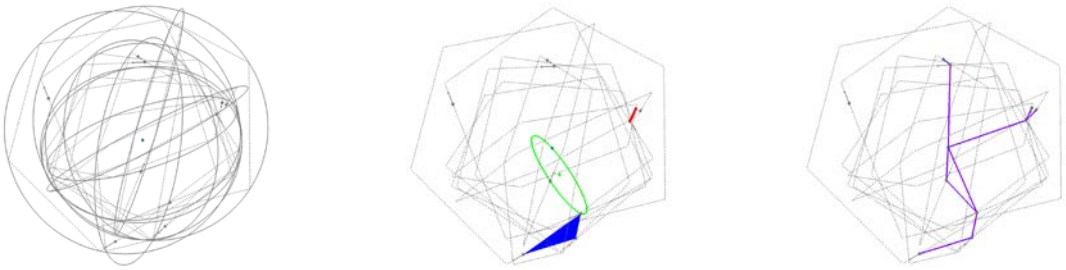
- Given the flipping history in the COOP-HSTRY file, it takes $O(k + N)$ time for constructing the Voronoi diagram at an arbitrary moment t^* where there are k flipping events before t^*.
- This proves that the proposed algorithm is event-based, efficiently replay-able. By definition, Voronoi diagrams are general-purpose and independent of coordinate system.

4. DVD-COOP: ANALYSIS AND VISUALIZATION

The DVD-COOP algorithm/program has powerful capabilities in spatial reasoning accurately, efficiently, and conveniently.

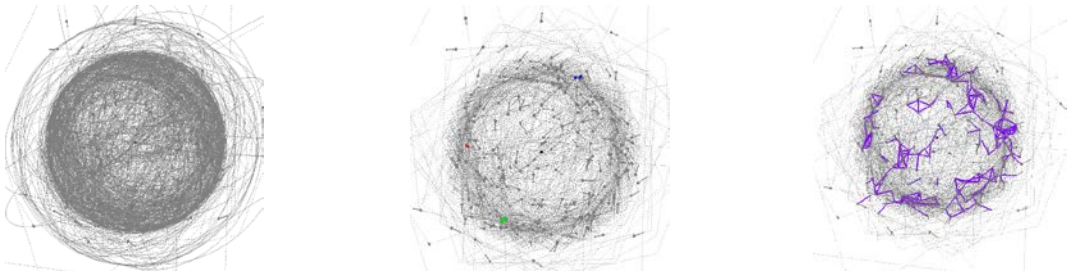
4.1. Proximity analysis

Fig. 5(a) shows the replicas of 10 geospace objects in Space Catalogue at an arbitrary moment, their orbits, and the piecewise linear approximations with five line segments. Recall that in this paper an object follows an elliptic orbit whereas an object replica moves through a linear approximated path. Fig. 5(b) shows the closest replica pair (the red line segment) which defines the shortest distance between two replicas, the closest triplet which defines the minimum radius circle (the green circle), and the minimal triplet which defines the minimum area triangle (the blue triangle). Fig. 5(c) shows the proximal pairs whose distances are within a threshold, in this case 6,000 km. Given the Voronoi diagram of the moment of interest in the prediction time window, all answers of the queries in Fig. 5(b) and (c) can be produced in $O(M)$ time in the worst case where M is the number of entities (i.e. V-vertices, V-edges, and V-faces) of the Voronoi diagram. For the space objects, $M = O(N)$ on average where N represents the number of replicas. Fig. 6 shows the same analyses as Fig. 5 but with 250 replicas: Fig. 6(c) shows the proximal pairs within 1,500 km. Fig 7 and 8 correspond to Fig 5 and 6 with the orbit resolution level of 20 line segments, respectively. We observe that the outputs are different depending on the resolution level. Be aware that the results in Fig. 5 through 8 are from object replicas, not from objects.



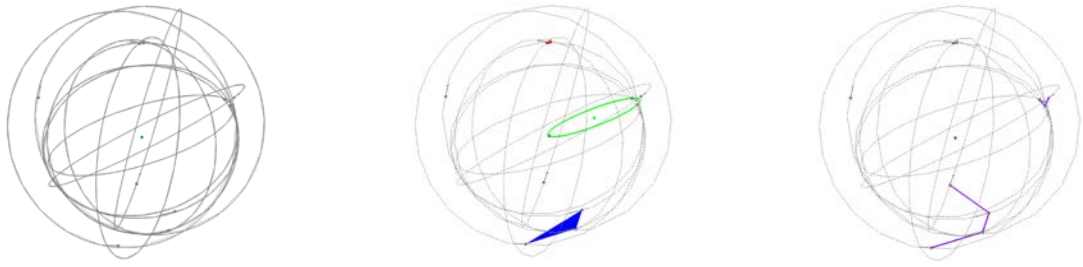
(a) (b) (c)

Fig. 5. An example of proximity analysis of DVD-COOP. Ten object replicas. Orbit approximation with five line segments. Object replicas follow approximated piecewise linear paths. (a) Orbits and their linear approximations. (b) Object replicas: The closest pair (the red line segment), the closest triplet (which defines the minimum radius circle; the green circle), and the minimal triplet (which defines the minimum area triangle; the blue triangle). (c) Object replicas: The proximal pairs whose distances are within 6,000 km.



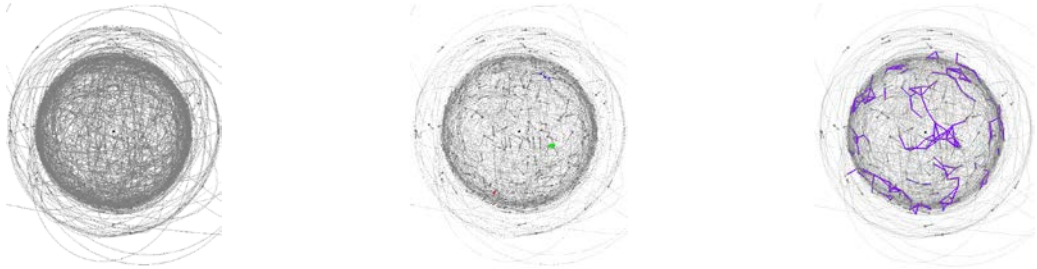
(a) (b) (c)

Fig. 6. Same as Fig. 5 with 250 object replicas. (a) Object replicas and the approximations of the orbits. (b) Proximity analysis as Fig. 5. (c) The pairs within 1,500 km.



(a) (b) (c)

Fig. 7. Repeat of Fig. 5 with the orbit approximation with 20 line segments. The output is different from Fig. 5.



(a) (b) (c)

Fig. 8. Repeat of Fig. 6 with the orbit approximation with 20 line segments. The output is different from Fig. 6.

4.2. Proximity analysis over the entire prediction time window

Fig. 9 shows the distances of 10,000 replica pairs with short distances over prediction time window of 30 seconds. The red curve corresponds to the distance between two closest replicas; The blue one for the replica pair with the second shortest distance; The green one for the third shortest distance. Fig. 9(a) and (b) correspond to the resolution levels of five and one thousand line segments, respectively. The three curve values at an arbitrary moment can be all computed in $O(M)$ time for M Voronoi entities if the Voronoi diagram is available. Similar analyses can also be done easily for the closest triplet, the minimal triplet, etc., all in $O(M)$ time. Recall that $M = O(N)$ on average for N replicas. We were at the moment not able to run any bigger data set and/or any longer prediction time window due to technical difficulties in implementation.

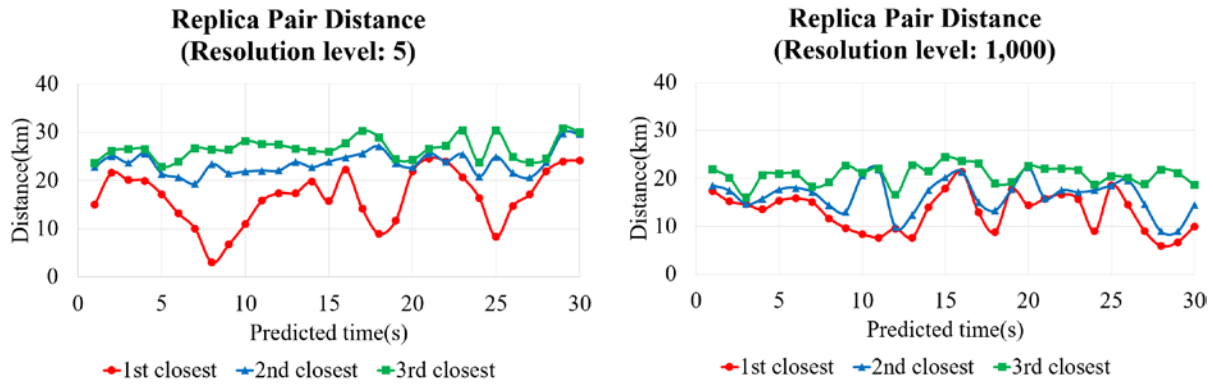


Fig. 9. Distances between replicas in the neighbor. 10,000 replicas of objects from Space Catalogue. 30 seconds prediction time window. (Red: The closest pair; Blue: The second closest pair; Green: The third closest pair). (a) Approximation with 5 line segments. (b) Approximation with 1,000 line segments.

5. GUARANTEED PERFORMANCE OF THE DVD-COOP ALGORITHM

The DVD-COOP algorithm guarantees to include the globally shortest distance pair of orbit objects in the reduced replica pair set. Therefore, if such an object pair defines a conjunction, the algorithm guarantees to find it with a small size of search space. In addition, it can be easily and effectively parallelized.

5.1. Guaranteed conjunction solution

Let O_A and O_B be two objects on the respective orbits and consider two consecutive moments t_i and t_{i+1} in the prediction time window (Fig. 10). Let L be the line segment connecting the two points O_i and O_{i+1} corresponding to the two moments on the orbit of O_A (i.e. $O_i = O_{A,i} = O_A(t_i)$,). Suppose that a replica of O_A, say $\hat{O}_A$, is moving through L (= L_A) during the time interval $\Delta t = t_{i+1} - t_i$ with a constant speed.

Let ε_A be the positional error of the replica $\hat{O}_A$ from the orbit object O_A. Suppose that Δt is small. We model the elliptic orbit corresponding to Δt with a circular arc: $r(t_i) = r(t_{i+1})$ where $r(t_i)$ represents the distance of O_A from the earth at t_i. We further model as O_A moves at a constant angular velocity: $\theta(t) - \theta(t_i) = 2\omega(t - t_i)$, $0 < \omega \in R$, where $\theta(t)$ is the true anomaly at t (i.e. the angle between the orbit object and the direction of periapsis as seen from the earth). $\hat{O}_A$ moves linearly at a constant speed.

For notational convenience, let $O = O(t)$, $\hat{O} = \hat{O}(t)$, and $r = r(t_i) = r(t_{i+1})$. Let $T = t - t_i$ be the elapsed time from t_i and $\Theta = \theta(t) - \theta(t_i)$ be the angular displacement of O from t_i. Let $\Theta/2 = \omega T$ for some constant ω as Θ is linear to T (explained above). Then, the positional error $\varepsilon = \varepsilon_A$ is given by the law of cosine as

$$\varepsilon^2 = l^2 + q^2 - 2lq\cos\beta \qquad (1)$$

from the triangle $\triangle O_i O\hat{O}$. Let l be the displacement of $\hat{O}$ from O_i, q the distance between O and O_i, and $\beta = \angle O O_i \hat{O}$. Then,

$$l = |L|T/\Delta t \quad (2)$$

where $|L|$ is the length of L. Note that $\hat{O}$ moves along L with a constant speed. From the isosceles $\triangle O_i O F$, the following equations are obtained.

$$q = 2r \sin\frac{\Theta}{2} = 2r \sin \omega T \quad (3)$$

$$\beta = \frac{\pi-\Theta}{2} - \alpha = \frac{\pi}{2} - (\omega T + \alpha) \quad (4)$$

where $\alpha = \angle \hat{O} O_i F$ and do not change between t_i and t_{i+1}. Substituting Eq.s (2), (3), and (4) into Eq. (1) produces

$$\varepsilon^2 = l^2 + q^2 - 2lq\cos\beta = l^2 + q^2 - 2lq\sin(\omega T + \alpha) = \frac{|L|^2 T^2}{\Delta t^2} + 4r^2 \sin^2 \omega T - \frac{4|L|Tr \sin \omega T \sin(\omega T + \alpha)}{\Delta t} \quad (5)$$

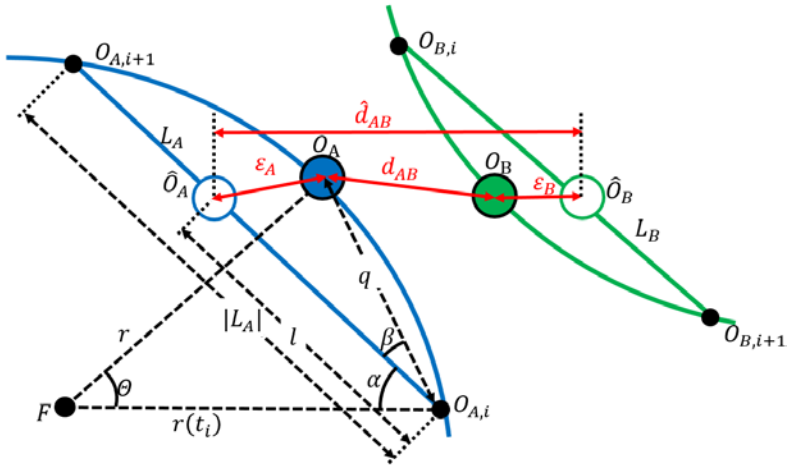


Fig. 10. Positional error between an orbit object and its replica at an arbitrary moment $t_i \le \hat{t} \le t_{i+1}$. The elliptic orbit is approximated by a circular arc with a constant angular velocity for a small time interval.

We verified Eq. (5) by calculating the true positional errors of a linearly moving replicas from the corresponding objects on their elliptic orbit in Space Catalogue. Orbits are approximated by ten line segments in the way that traveling time between consecutive vertices are identical. Fig. 11(a) shows the experimental result of the object INS-1A (Eccentricity: 0.001; Semi-major axis: 6,878 km). The red circles are the calculated errors *CalcError* by Eq. (5) which are almost identical to the real errors *RealError* (thus hidden by the red circles). The green curve corresponds to *CalcError – RealError*. For each line segment approximation, we evaluated 100 sample points with a constant time increment. As $r(t_i) \neq r(t_{i+1})$ on the elliptic orbit, we used $r = (r(t_i) + r(t_{i+1}))/2$ for the circle approximation. Note that the calculated, predicted positional error by Eq. (5) in this case is very close to the real one only with ten line segment approximation.

Fig. 11(b) and (c) show the same experiments with the objects BREEZE-M R/B (Eccentricity: 0.05; Semi-major axis: 45,461 km) and DELTA 1 R/B(1) (Eccentricity: 0.2; Semi-major axis: 8,642 km), respectively. The blue circles denote the real errors. We observe that the deviation of predicted positional error increases as orbit eccentricity increases. In Fig. 11(c), the largest deviation is even about 100 km: Recall that we approximated the orbits with just ten line segments! Note that the green curve behaves in a bit weird fashion at the common locations between elliptic orbit and its piecewise linear approximation. This is because we assumed that $r = r(t_i) = r(t_{i+1})$ in the error formula in Eq. (5). We also observe that the positional error decreases as the true anomaly approaches 180 degree. Fig. 11(d) shows the profile of orbit eccentricity in the entire Space Catalogue.

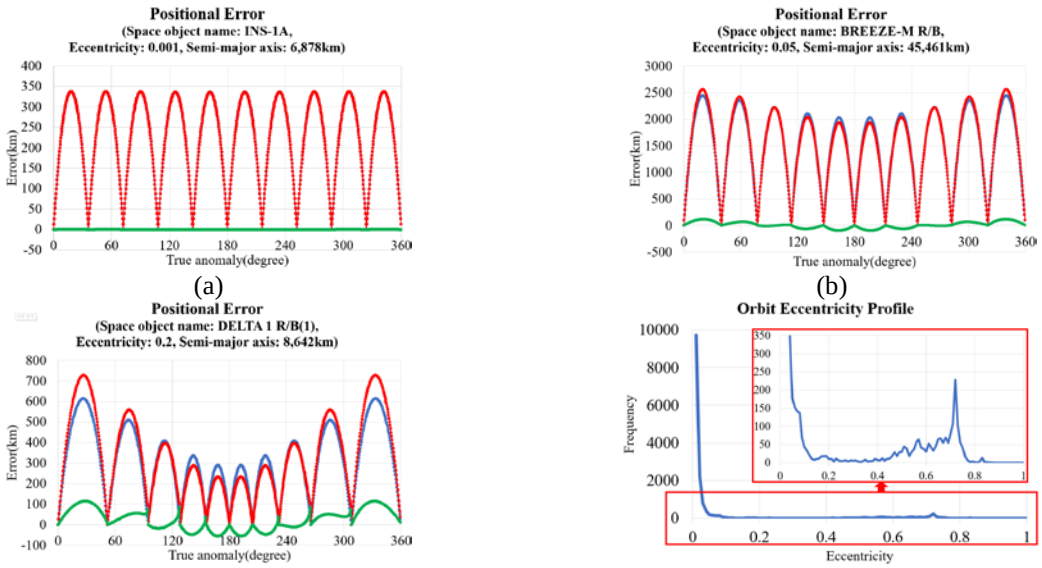


Fig. 11. The real and calculated positional errors. Red: calcualted error; Blue: real error; Green: Red – Blue. (a) Space object INS-1A (Eccentricity: 0.001; Semi-major axis: 6,878 km), (b) BREEZE-M R/B (Eccentricity: 0.05; Semi-major axis: 45,461 km), (c) DELTA 1 R/B(1) (Eccentricity: 0.2; Semi-major axis: 8,642 km), (d) The profile of the eccentricity of the orbits in Space Catalogue.

In a similar fashion, L_{B} and $\hat{O}_B$ of the orbital object O_{B} can be defined together with the positional error ε_B. Recall that the replicas move through piecewise linear paths with constant speeds while objects move through their respective elliptic orbits under Keplerian motion.

Suppose that the true shortest distance d^*_{AB} between O_{A} and O_{B} occurs at $t_{\mathrm{i}} \le t^* \le t_{\mathrm{i+1}}$ at the locations, say O^*_{A} and O^*_{B}, on their respective orbits. Suppose that $\hat{O}_A$ and $\hat{O}_B$ are identified, from the Voronoi diagram, as the closest pair among all possible replica pairs at $t_{\mathrm{i}} \le \hat{t} \le t_{\mathrm{i+1}}$ (Fig. 11). Note that this closest pair between replicas can be found from the Voronoi diagram by checking the V-faces, thus taking O(M) time for M Voronoi faces where M = O(N) on average for our problem for N orbital objects. It is important to note that t^* is not necessarily identical to $\hat{t}$.

Let $\hat{d}_{AB}$ be the distance between the centers of the replicas $\hat{O}_A$ and $\hat{O}_B$ at $\hat{t}$. Then, it is obvious that the shortest distance d^*_{AB} between two elliptic orbits satisfies

$$\hat{d}_{AB} - (\varepsilon_A + \varepsilon_B) \le d^*_{AB} \le \hat{d}_{AB} + (\varepsilon_A + \varepsilon_B). \quad (6)$$

Let $\varepsilon = \max(\varepsilon_{\mathrm{A}}, \varepsilon_B)$. Then,

$$\hat{d}_{AB} - 2\varepsilon \le d^*_{AB} \le \hat{d}_{AB} + 2\varepsilon. \quad (7)$$

Eq. (6) and (7) imply that the replica pair $\hat{O}_A$ and $\hat{O}_B$ with the shortest distance $\hat{d}_{AB}$ on the linear approximations may not necessarily correspond to the object pairs with the shortest distance d^*_{AB} among all objects on orbits. The same equation, however, also implies that we can exclude to filter out any replica pair $\hat{O}_{\mathrm{i}}$ and $\hat{O}_{\mathrm{j}}$ which satisfies the condition

$$|\hat{d}_{IJ} - \hat{d}_{AB}| \ge 2\varepsilon \quad (8)$$

from the consideration. The globally shortest distance pair of objects on the orbits can be correctly found from the set of replica pairs after the filter in Eq. (8) is applied. Let X be the set of replica pairs, called the **reduced replica pair set**, after the filtering process. It can be shown that t^*, O^*_{A}, O^*_{B}, and d^*_{AB} can be found by applying numerical procedure to each reduced replica pair ($\hat{O}_A$, $\hat{O}_B$) using $\hat{t}$ as an initial condition. We make the following observations:

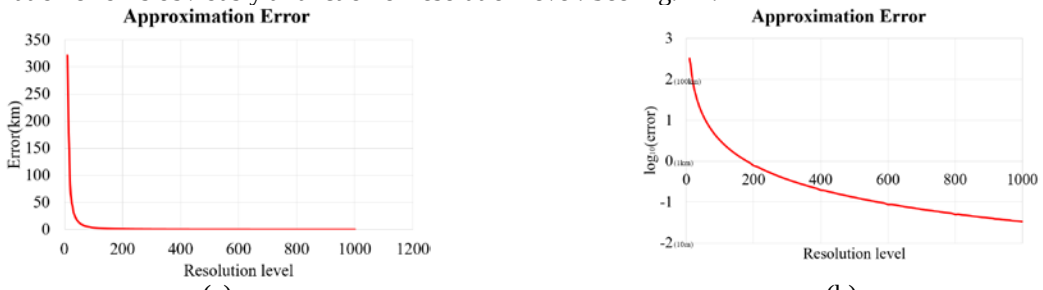
- (**Obs 1. Existence of Solution**) The shortest distance object pair exists in the reduced replica pair set X.
- (**Obs 2. Size of Search Space**) The value of ε and the size of X have a strong positive correlation.

5.2. Size of search space

The history of the events (i.e. V-edge flip and V-face flip events) during the entire prediction time window is stored in the COOP-HSTRY file. Given a COOP-HSTRY file, the topology of the Voronoi diagram of any arbitrary moment, say t, in the conjunction prediction time window can be quickly computed in the linear time of the number k of events up to t and the geometry of the Voronoi diagram $VD(t)$ at t can be computed in the linear time of the number N of object replicas. The DVD-COOP program can be easily modified to efficiently solve many other useful spatial queries. Two examples:

- Suppose that a user wants to identify the neighbor replicas of a particular replica, say $\hat{O}$, within a threshold τ at t. Then, locating the replica of the object O in $VD(t)$ and checking the V-faces of the V-cell of $\hat{O}$ can solve the problem. Given $VD(t)$, this query can be done in O(N) time.
- Suppose that a user wants to see the distance of the nearest neighbor of a particular replica $\hat{O}$ over the entire prediction time window. This can also be done quickly by scanning the entire COOP-HSTRY file by performing the flipping events and checking the V-faces of the V-cell corresponding to $\hat{O}$. This query can be done in $O(k)$ time.

Approximation error is obviously a function of resolution level. See Fig. 12.



(a) (b)

Fig. 12. Approximation error with respect to the number of approximating line segments of an orbit. (a) Absolute error. (b) Log-scale error.

The resolution level eventually transforms to computational requirement, firstly, to find conjunctions among replicas and secondly, to find the conjunctions among real orbital objects. Fig. 13(a) shows the computation time to find the conjunctions among replicas, not among orbital objects, including the computation of the Voronoi diagram of both the initial and the dynamic over the entire prediction time window. Note that the curve in Fig. 13(a) increases in a strongly linear fashion. This is because there are more velocity vector changes as there are more approximating line segments for orbits during the prediction time. When a velocity vector is modified, to maintain a correct Voronoi diagram, it is necessary to update the V-flipping time of the V-edges and V-faces in the neighborhood of the velocity-changed replica. The number of velocity changes is obviously linear to the number of the line segments.

The search space size observation (Obs 2) above says that if we approximate two orbits with more line segments, we need to check a smaller set of reduced replicas which guarantees no missing conjunctions among orbital objects. Fig. 13(b) shows the size of the reduced replica pair set which contains the shortest distance (i.e. the real conjunction) among orbital objects as the consequence of Obs 2 above. As the resolution level gets higher (i.e. the approximation error becomes smaller), there are fewer replica pairs from which the real conjunction should be tested. However, Fig. 13(a) says that the computation to get the reduced replica set takes more time for resolution level increase. Fig. 13(a) and (b) together yield a trade-off which implies the existence of an optimal condition of resolution level which can be determined by a computational experiment conjunction problem. The optimality may depend on platform. Thus, we make the following observation.

- (**Obs 3. Optimality of Efficiency**) There exists an optimal resolution level for the minimum computation time in conjunction prediction.

(a) (b)

Fig. 13. The influence of approximation quality by resolution level (10,000 objects from Space Catalogue, 30 seconds prediction time window, Intel Core i7-6700 3.4Ghz, 16Gb RAM, Win10 64bit, one core used). An orbit approximation resolution level is the number of line segments approximating an orbit. (a) Computation time of both the initial Voronoi diagram and the dynamic Voronoi diagram (i.e. the V-flip history) over the entire conjunction prediction time window. (b) The number of reduced replica pairs.

5.3. Scalability of DVD-COOP:

The experimental results above were produced by the DVD-COOP algorithm running on a single core and the computation speed can be significantly improved if the following three types of scalability are properly incorporated. We plan to implement the scalable version of the proposed algorithm soon.

- Prediction-window scalability (PW-scalability): PW-scalability is based on the observation that the prediction time window can split into a set of mutually exclusive time slices and the conjunction prediction of each time slice can be done independent of each other. Thus, the predictions of all time slices can be done in parallel, if an actual collision does not occur between two objects in a preceding time slice. Note that the beginning configuration of each time slice can be easily found. This type of scalability has been commonly used by other existing algorithms/programs.
- Flip-influence scalability (FI-scalability): FI-scalability is unique in the DVD-COOP algorithm. When a V-edge (or a V-face) flips in the topology structure of the Voronoi diagram, there are a number of V-edges and V-faces whose flipping event time in future need to be re-calculated and these re-calculations are independent of each other. Hence, these re-calculations can be done in parallel. According to our current experiment, the number of re-calculations per flip is 46 on average and 64 in the worst case. These numbers refer to the possible scale of the speed-up due to FI-scalability. As the largest portion of the computation time of DVD-COOP is this re-calculation of event time, the FI-scalability will easily achieve 46 times acceleration if it is appropriately reflected in algorithm design.
- Neighbor-pair scalability (NP-scalability): Given two replicas of each of the reduced replica pair set, the minimum distance between the corresponding orbit objects can be computed independent of other replica pairs. NP-scalability is based on this observation. As there are usually many elements in the reduced replica pair set, the NP-scalability should be very effective.

6. DISCUSSIONS AND CONCLUSION

The proposed DVD-COOP algorithm based on the dynamic Voronoi diagram is general-purpose, accurate, efficient for the conjunction prediction of space situational awareness problems. The proposed algorithm can be easily used to answer to many spatial queries about moving objects including conjunction prediction. The algorithm is based on the dynamic Voronoi diagram of 3D spherical objects moving through respective orbits. The DVD-COOP program currently implements the proposed algorithm in the standard C++ language and uses only one core. The experiment statistics were produced using a subset of Space Catalogue of JSpOC database due to the technical limitation in the current implementation. We hope the idea presented in this paper could transform the paradigm for computational prediction of space situations.

There are many remaining issues to be solved in DVD-COOP: Parallelization, securing numerical robustness, parameter optimization, code optimization, etc. Many of these technical issues are challenging in that they touch the boundary of theoretically unsolved problems. Incorporation of the uncertainty to the nominal geometry of the current model is another critical challenge.

We envision that the COOP programs might be used in the configuration shown in Fig. 14. When the JSpOC Space Catalogue is updated, the DVD-COOP algorithm/program is executed to produce an event history over a conjunction prediction time window and store it in the COOP-HSTRY file. The COOP-HSTRY file might also be stored in JSpOC database to be downloaded by users. Individual user then runs an analysis program such as COOP-ANALYZER to analyze COOP-HSTRY according to individual need.

We are committed to develop and distribute these COOP programs. We also plan to develop a library based on the proposed algorithm: The one like STK(Systems Tool Kit; previously Satellite Tool Kit) by Analytical Graphics, Inc. (AGI, http://www.agi.com/) whose APIs were used for easy development of application programs: E.g. see [49].

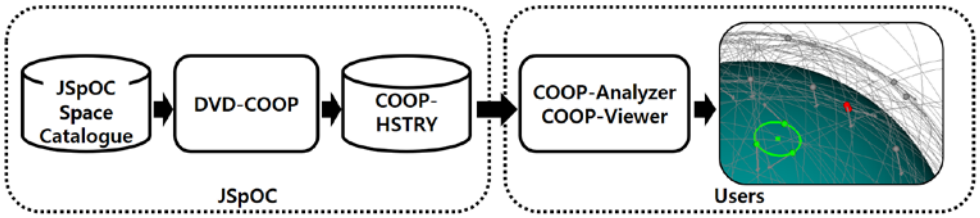


Fig. 14. A possible configuration of DVD-COOP uses

7. REFERENCES

1. Rising, D., *Satellite Hits Atlantic—but What about Next One?*, Seattle Times, 2013.
2. Global experts Agree Action Needed on Space Debris, European Space Agency, 2013. http://www.esa.int/Our_Activities/Operations/Space_Debris/Global_experts_agree_action_needed_on_space_debris
3. How many space debris objects are currently in orbit?, European Space Agency, 2013. http://www.esa.int/Our_Activities/Space_Engineering_Technology/Clean_Space/How_many_space_debris_objects_are_currently_in_orbit
4. Budianto-Ho, I.A. et al, Scalable Conjunction Processing using Spatiotemporally Indexed Ephemeris Data, In *Advanced Maui Optical and Space Surveillance Technologies Conference*, 2014.
5. Liou, J.C. and Johnson, N.L., Risks in Space from Orbiting Debris, *Science*, Vol. 311, 340-341, 2006.
6. Sabol, C. et al, Linearized Orbit Covariance Generation and Propagation Analysis via Simple Monte Carlo Simulations, In *Proceedings of the 2010 Space Flight Mechanics Conference*, 2010.
7. George, E. and Harvey, S., A Comparison of Satellite Conjunction Analysis Screening Tools, In *Advanced Maui Optical and Space Surveillance Technologies Conference*, 2011.
8. Alarcón-Rodríguez, J.R. et al, Conjunction event predictions for operational ESA satellites, In *Proceedings of the Space-Ops 2004 Conference*, 2004.
9. Alarcón-Rodrıguez, J.R. et al, Development of a collision risk assessment tool, *Advances in Space Research*, Vol. 34, 1120-1124, 2004.
10. Casanova, D. et al, Space Debris Collision Avoidance Using a Three-filter Sequence, *Monthly Notices of the Royal Astronomical Society*, Vol. 442, 3265-3242, 2014.
11. Wisniowski, T. and Rickman, H., Fast Geometric Method for Calculating Accurate Minimum Orbit Intersection Distances, *Acta Astronomica*, Vol. 63, 293-307, 2013.
12. Chen, H. et al, Conformal Prediction for Anomaly Detection and Collision Alert in Space Surveillance, In *Proceedings of SPIE 8739*, 2013.
13. Alfano, S., Toroidal Path Filter for Orbital Conjunction Screening, *Celestial Mechanics and Dynamical Astronomy*, Vol. 113, 321-334, 2012.
14. Crassidis, J. et al, Space Collision Avoidance, In *Proceedings of National Symposium on Sensor and Data Fusion*, 2011.
15. Armellin, R. et al, Computing the Critical Points of the Distance Function between Two Keplerian Orbits via Rigorous Global Optimization, *Celestial Mechanics and Dynamical Astronomy* Vol. 107, 377-395, 2010.
16. Woodburn, J. et al, A Description of Filters for Minimizing the Time Required for Orbital Conjunction Computations, *Advances in the Astronautical Sciences*, Vol. 135, 1157-1173, 2009.
17. Healy, L.M., Close Conjunction Detection on Parallel Computer, *Journal of Guidance, Control, and Dynamics*, Vol. 18, 824-829, 1995.
18. Coppola, V. and Woodburn, J., Determination of Close Approaches Based on Ellipsoidal Threat Volumes, *Advances in the Astronomical Sciences: Spaceflight Mechanics*, Vol. 102 1013-1024, 1999.

19. Carlson, R. and Lee, J., Detecting Near Collisions for Satellites, *IEEE transactions on aerospace and electronic systems*, Vol. 33, 921-929, 1997.
20. Hoots, F.R. et al, An Analytic Method to Determine Future Close Approaches between Satellites, *Celestial mechanics*, Vol. 33, 143-158, 1984.
21. Kubica, J. et al, Efficient Intra-and Inter-night Linking of Asteroid Detections Using Kd-trees, *Icarus*, Vol. 189, 151-168, 2007.
22. Kouprianov, V., ISON Data Acquisition and Analysis Software. In *6th European Conference on Space Debris*, 2013.
23. Denneau, L. et al, The Pan-STARRS Moving Object Processing System. *Publications of the Astronomical Society of the Pacific*, Vol. 125, 357, 2013
24. Mercurio, M. et al, A Hierarchical Tree Code Based Approach For Efficient Conjunction Analysis, In *AIAA/AAS Astrodynamics Specialist Conference*, 2012.
25. Bentley, J.L., Multidimensional Binary Search Trees Used for Associative Searching., *Communications of the ACM*, Vol. 18, 509-517, 1975.
26. Alarcón-Rodriguez, J.R. et al, Collision Risk Assessment With a "Smart Sieve" Method, In *Joint ESA-NASA Space-Flight Safety Conference*, 2002.
27. Okabe, A. et al, *Spatial Tessellations - Concepts and Applications of Voronoi Diagram*, John Wiley & Sons, 1992.
28. Kim, D.-S. et al, Voronoi Diagram of a Circle Set from Voronoi Diagram of a Point Set: I. Topology, *Computer Aided Geometric Design*, Vol. 18, 541-562, 2001.
29. Kim, D.-S. et al, Voronoi Diagram of a Circle Set from Voronoi Diagram of a Point Set: II. Geometry, *Computer Aided Geometric Design*, Vol. 18, 563-585, 2001.
30. Kim, D.-S. et al, Euclidean Voronoi diagram of 3D Balls and Its Computation Via Tracing Edges, *Computer-Aided Design*, Vol. 37, 1412-1424, 2005.
31. Kim, D. and Kim, D.-S., Region-Expansion for the Voronoi Diagram of 3D Spheres, *Computer-Aided Design*, Vol. 38, 417-430, 2006.
32. Kim, D.-S. et al, Quasi-Triangulation and Interworld Data Structure in Three Dimensions, *Computer-Aided Design*, Vol. 38, 808-819, 2006.
33. Kim, D.-S. et al, Three-Dimensional Beta Shapes, *Computer-Aided Design*, Vol. 38, 1179-1191, 2006.
34. Kim, D.-S. et al, Quasi-worlds and Quasi-operators on Quasi-triangulations, *Computer-Aided Design*, Vol. 42, 874-888, 2010.
35. Kim, D.-S. et al, Three-dimensional Beta-shapes and Beta-complexes via Quasi-triangulation, *Computer-Aided Design*, Vol. 42, 911-929, 2010.
36. Kim, D.-S. et al, Querying Simplexes in Quasi-triangulation, *Computer-Aided Design*, Vol. 44, 85-98, 2012.
37. Kim, D.-S. et al, QTF: Quasi-triangulation File Format, *Computer-Aided Design*, Vol. 44, 835-845, 2012.
38. Lee, M. et al, Topology-oriented Incremental Algorithm for the Robust Construction of the Voronoi Diagrams of Disks, *ACM Transactions on Mathematical Software*, Vol. 43, 14, 2016.
39. Kim, D.-S. et al, Euclidean Voronoi Diagrams of 3D Spheres: Their Construction and Related Problems from Biochemistry, *Lecture Notes in Computer Science*, Vol. 3604, 255-271, 2005.
40. Kim, D.-S. et al, Crystal Structure Extraction in Materials using Euclidean Voronoi Diagram and Angular Distributions among Atoms, *Journal of Ceramic Processing Research*, Vol. 6, 63-67, 2005.
41. Ryu, J. et al, Molecular Surfaces on Proteins via Beta Shapes, *Computer-Aided Design*, Vol. 39, 1042-1057, 2007.
42. Kim, D.-S. et al, BetaDock: Shape-priority Docking Method Based on Beta-complex, *Journal of Biomolecular Structure & Dynamics*, Vol. 29, 219-242, 2011.
43. Kim, D.-S. et al, Beta-decomposition for the Volume and Area of the Union of Three-dimensional Balls and Their Offsets, *Journal of Computational Chemistry*, Vol. 33, 1252-1273, 2012.
44. Shin, W.-H. et al, GalaxyDock2: Protein-Ligand Docking using Beta-Complex and Global Optimization, *Journal of Computational Chemistry*, Vol. 34, 2647-2656, 2013.
45. Kim, J.-K. et al, BetaCavityWeb: a Webserver for Molecular Voids and Channels, *Nucleic Acids Research*, Vol. 32, W413-W418, 2015.
46. Park, C.H. et al, Nanocrack-Regulated Self-Humidifying Membranes, *Nature*, Vol. 532, 480-496, 2016
47. Gavrilova, M. and Rokne, J., Swap Conditions for Dynamic Voronoi Diagrams for Circles and Line Segments, *Computer Aided Geometric Design*, Vol. 16, 89-106, 1999.
48. Gavrilova, M. and Rokne, J., Updating the Topology of the Dynamic Voronoi Diagram for Spheres in Euclidean D-dimensional Space, *Computer Aided Geometric Design*, Vol. 20, 231-242, 2003.
49. Kim, H.-D. et al, Analysis of Space Debris Collision Risk using KARISMA for KOMPSAT Satellites Series, In *SpaceOps 2014 Conference*, 2014.