DVD-COOP for Maneuver Path Optimization of Conjunctive Resident Space Objects for Space Traffic Management

Jehyun Cha[1], Drew McNeely[2], Joonghyun Ryu[1], Misoon Mah[3], Moriba Jah[2], and Deok-Soo Kim[1,*]

[1] School of Mechanical Engineering, Hanyang University, Seoul, Korea
[2] Aerospace Engineering & Engineering Mechanics Department, The University of Texas at Austin, Texas, USA
[3] US Air Force Office of Scientific Research (AFOSR), USA
** Corresponding author(dskim@hanyang.ac.kr)*

Abstract

Collision avoidance is one of the critical tasks to keep geospace safe and efficient. The detection and resolution of conjunction is a prerequisite for collision avoidance. In AMOS 2017, we reported a study of the DVD-COOP algorithm/program, based on the dynamic Voronoi diagram 3D spheres, which showed a proven mathematical capability together with implementation to detect all conjunctions without any missing case. In this paper, we report a new capability of the DVD-COOP algorithm which, given a detected conjunction, produces the optimal maneuver plan by quickly evaluating alternatives using the event history stored in the COOP-HSTRY file. The DVD-COOP algorithm, with a strong scalability, seems to play a key role for Space Situational Awareness and Space Traffic Management.

1. Introduction

There are many Resident Space Objects (RSOs) in geospace. As of 2013, more than 6,600 satellites have been launched since 1957: 3,600 remained in orbit [1] including about 1,000 operational ones [2]. European Space Agency (ESA) reported approximately 29,000 debris of size bigger than 10 cm, 670,000 debris of size 1~10 cm, more than 170 million ones smaller than 1 cm [3], as of July 2013.

This number will increase rapidly due to accidental satellite collisions (e.g. 2009 collision between Iridium 33 and Kosmos 2251; Produced >2,000 catalogued debris), planned anti-satellite missile test (e.g. 2007 Chinese missile test against Fungyun satellite; Produced >2,000 trackable ones among 150,000 debris), new satellite launches, etc. Attention should also be provided for the increase of the number of deployed small satellites in geospace. In recent years, the trend of deployment of spacecrafts is more, smaller, and lower cost civilian ones compared to the few, large, and expensive government ones [4]. These altogether significantly increases collision risks [5]. As RSOs move at high speed of up to 16 km/s if head-on, the impact of collision between space objects can be catastrophic.

To better prevent unexpected collisions among objects and to preserve the geospace, particularly Low Earth Orbit (LEO), for future, it is necessary to have a method to predict and prevent collisions and eventually develop space traffic management (STM) system. STM will be increasingly important as the geospace is more commercialized such as the expected popularity of suborbital space tourism and/or commercial personal spaceflight [6, 7].

One of the most critical issues in STM is to predict and prevent collisions among RSOs [8, 9]. Hence, the detection, tracking, identification, cataloging, etc. of all observable RSOs in the orbit, altogether referred to as space situational awareness (SSA), is necessary. Securing a perfect SSA is both costly and complicated. Joint Space Operations Center (JSpOC, https://www.space-track.org/) maintained by the United States Strategic Command's (USSTRATCOM) is a good resource [10]. JSpOC contains 17,043 RSOs as of August 22, 2018 and is expected soon to contain more than 100,000 [11, 12]. One of the critical SSA issues is the prediction of conjunction, i.e. a dangerous condition of co-located objects within a near proximal distance, in which case the probability of collision rapidly increases.

Accurate and efficient computation for conjunction prediction is critical. Hence, there are many studies. Most existing studies were about the reduction of search space by filtering out orbit combinations that were guaranteed to be free of any conjunction [13, 14, 15, 16, 17, 18, 19, 20, 21, 22] which were the improvements of the initial idea of the three-filter approach proposed by Hoots et al. (1984) [23]: Apogee/perigee filter, orbit path filter, and time filter. Recently,

the Moving Object Processing System (MOPS) system of University of Hawaii's Pan-STARRS adapted kd-trees for the faster prediction of conjunctions in 2002 [24, 25 26, 27, 28]. CAOS-D system also employed the kd-tree for conjunction prediction [10].

The value of conjunction prediction is obvious in that, if a conjunction is correctly predicted, a collision-avoiding evasive maneuver of a RSO can be planned and executed. When it is possible, an optimal maneuver pathway might need to be determined or designed by possibly evaluating the effect of each hypothesized maneuver 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 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 [11, 12].

In this paper we propose a new capability of the DVD-COOP program to find the optimal maneuver pathway for a predicted conjunction situation among the RSOs 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.

2. Dynamic Voronoi Diagram

In this section, we provide a brief introduction of the Voronoi diagram as it is the core of the proposed study. The **ordinary Voronoi diagram** *Vor*(*P*) of a point set *P* is a tessellation or tiling 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 [29, 30]. Given a Voronoi diagram, spatial queries can be correctly and efficiently (or at least most accurately) answered. The **Voronoi diagram** *VD*(*B*) of a spherical ball set *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. For VD in 3D, 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 [31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41]. Queries about spatial reasoning among spherical objects can be easily solved with VD. **Dynamic Voronoi diagram (DVD)** is the Voronoi diagram for moving particles. Hence, if a dynamic Voronoi diagram is available, the powerful spatial reasoning algorithms can be applied to moving objects. Given the initial Voronoi diagram and the input particle information (i.e. locations and velocities), the DVD algorithm detects the moments where critical events for maintaining a correct Voronoi diagram structure occurs. In three-dimensional space, the flipping moment from a V-edge to a V-face, or vice versa, is a root of 10-th degree polynomial [42, 43]. We store the predicted history of consecutive flipping events in the **COOP-HSTRY file** taking $O(k)$ memory for k flipping events during the prediction time window.

3. Conjunction Prediction using DVD-COOP

The DVD-COOP algorithm/program can perform spatial reasoning among RSOs accurately, efficiently, and conveniently. When it runs the dynamic Voronoi diagram algorithm to predict future events, the DVD-COOP program produces the COOP-HSTRY data which stores the information about topology change in the Voronoi diagram of n RSOs during the prediction time. With the COOP-HSTRY data and the initial static Voronoi diagram at $t = 0$, the static Voronoi diagram for arbitrary time t can be computed in $O(|H| + n)$ time in the worst case where $|H|$ represents the size of COOP-HSTRY data before t. Hence, any spatial reasoning at t can be done by the efficiently computed static Voronoi diagram at that moment.

Among diverse applications of the COOP-HSTRY data, we are interested in the following two problems: Conjunction interval search problem (CIS-problem) and maneuver path optimization problem (MPO-problem). The CIS-problem is to find the conjunction interval between all possible pairs of RSOs. MPO-problem is to find the best conjunction-avoiding maneuver path by generating-and-testing through quick-and-accurate evaluation of multiple hypothesized paths.

Suppose that Space Catalogue has n RSOs (or equivalently n orbits) and thus O(n^2) RSO pairs. We make a linear approximation of each orbit with M_L line segments corresponding to some discrete moments in time. Consider a line segment L approximating an elliptic orbit. Let $\hat{O}$ be the replicas of an RSOs O: $\hat{O}$ move through L with a constant speed whereas O move through its elliptic orbit. Let ε be the positional error between O and $\hat{O}$. We take advantage of the fact that reasoning the spatial proximity among the linearly moving replicas can be easily done via the Voronoi diagram of the replicas.

3.1 Conjunction Interval Search (CIS) Problem

A conjunction interval is an interval within a prediction time window where the mean distance between any two RSOs is less than a predefined threshold θ. A conjunction interval contains one or more Time of Closest Approach (TCA) which gives a local minimum of the distance between any two RSOs. A CIS-problem is to find all RSO pairs and time interval where each has an inter-object distance less than θ. In a CIS-problem, being a building block of SSA/STM, we are interested in an efficient, convenient, and robust algorithm that finds all conjunction intervals within a prediction time window using the COOP-HSTRY data. We emphasize that the CIS-problem is one of the most fundamental geometric problems in SSA/STM.

Suppose that Space Catalogue is a set of n RSOs where each RSO follows an elliptic orbit, i.e. $O(t) = \{o_1(t), o_2(t), \dots, o_n(t)\}$ where $O(t)$ is the Catalogue. Consider a line segments L_i that approximates the elliptic orbit of o_i. Let $\hat{O}(t) = \{\hat{o}_1(t), \hat{o}_2(t), \dots, \hat{o}_n(t)\}$ denotes the replicas of an object $O(t)$. The object o_i moves through its elliptic orbit and $\hat{o}_i$ moves through L_i and has a positional error ε_i from o_i. The analysis for linear approximation of elliptic orbits and error estimation was discussed in our paper in AMOS 2017 [44].

Mathematically, the CIS-problem is to find all RSO pairs o_i and $o_j$where each pair is associated with a time interval t_{start} and t_{end} where

$$d_{ij}(\tau) \leq \theta \tag{1}$$

for the inter-object distance $d_{ij}(\tau)$ and $\tau \in [t_{start}, t_{end}]$. In other words, a CIS-problem is to find all $\{o_i, o_j, t_{start}, t_{end}\}$, called the **CI-quadruplet**, satisfying Eq. (1).

Let $\hat{d}_{ij}$ denotes the distance between two replicas $\hat{o}_i$ and $\hat{o}_j$. We showed in [44] that

$$\hat{d}_{ij} - (\varepsilon_i + \varepsilon_j) \leq d_{ij} \leq \hat{d}_{ij} + (\varepsilon_i + \varepsilon_j) \tag{2}$$

Eq. (2) implies that the true distance between RSOs can be bound by the distance between replicas if their positional errors are correctly estimated. From Eq. (1) and (2), the equation below is satisfied:

$$\hat{d}_{ij} - (\varepsilon_i + \varepsilon_j) \leq d_{ij} \leq \theta \tag{3}$$

Eq. (3) can be easily transformed to

$$\hat{d}_{ij} \leq \theta + (\varepsilon_i + \varepsilon_j). \tag{4}$$

Eq. (4) leads to the following Lemma.

Lemma 1. If $\hat{d}_{ij} \leq \theta + (\varepsilon_i + \varepsilon_j)$, then $d_{ij} \leq \theta$.

Lemma 1 implies that the proper distance threshold for replicas can be bounded with the original threshold and positional errors. Let $[\hat{t}_{start}, \hat{t}_{end}]$ be the conjunction interval for a replica pair satisfying Eq. (4). In other words, $\hat{d}_{ij}(\tau) \leq \theta + (\varepsilon_i + \varepsilon_j), \tau \in [\hat{t}_{start}, \hat{t}_{end}]$. From Lemma 1, it can be also shown that

$$[t_{start}, t_{end}] \subset [\hat{t}_{\text{start}}, \hat{t}_{end}]. \tag{5}$$

Eq. (5) implies that we are guaranteed not to miss any conjunction interval if we find the replica pairs satisfying Eq. (4) and closely investigate their corresponding orbital objects. Hence, the following theorem holds.

Theorem 2. The conjunction interval for replica always contains the conjunction interval for RSOs.

(Example: 2D) Fig. 1 shows an example of the process to track the inter-replica distance using the COOP-HSTRY data in the plane. Note that the figure illustrates a relative motion analysis. While we give the explanation in the plane, the observation easily applies to the three-dimensional space. All particles in the figure are fixed except the horizontally moving red one and we want to analyze the distance between the red and green. Suppose Fig. 1(a) is an initial state at time t_0: The red and green are not V-neighbors to each other which implies that they are not the first order neighbors in the Voronoi diagram (We call two objects V-neighbors if they share a Voronoi edge). In this case, we consider they do not define a distance.

Suppose the red moves linearly with a constant speed to the direction given by the arrow and also suppose that there exists $t_s > t_0$ such that four objects, including both the green and the red, define a common empty tangent circle as shown in Fig. 1(b). This implies that, at t_s, the red and green begin to be V-neighbors and thus the inter-object distance is defined. From the algorithm point of view, the Voronoi edge which shrinks to the point (corresponding to the center of the tangent circle) flips. The distance is well-defined until $t_f > t_s$ where another edge-flip occurs at t_f as shown in in Fig. 1(d). Note that the local closest approach occurs at t_{min} as shown in Fig. 1(c). The red and green are not V-neighbors any more after t_f and thus no distance is defined between the red and green at $t_\infty > t_f$ in Fig. 1(e). Fig. 1(f) shows the distance function between the red and green objects. For the relative position $p = \{p_1, p_2\}$ and velocity $v = \{v_1, v_2\}$, $p, v \in \mathbb{R}^2$, the distance d between linearly moving replicas with constant speeds is given as a square-rooted quadratic function

$$d^2 = \sum v_i^2 t^2 + 2\sum v_i p_i t + \sum p_i^2. \qquad (6)$$

Therefore, the following lemma holds.

Lemma 3. Suppose that there is no velocity change for two linearly moving objects. Then, the distance between the two objects consists of one, and only one, curve segment of the function in Eq. (6).

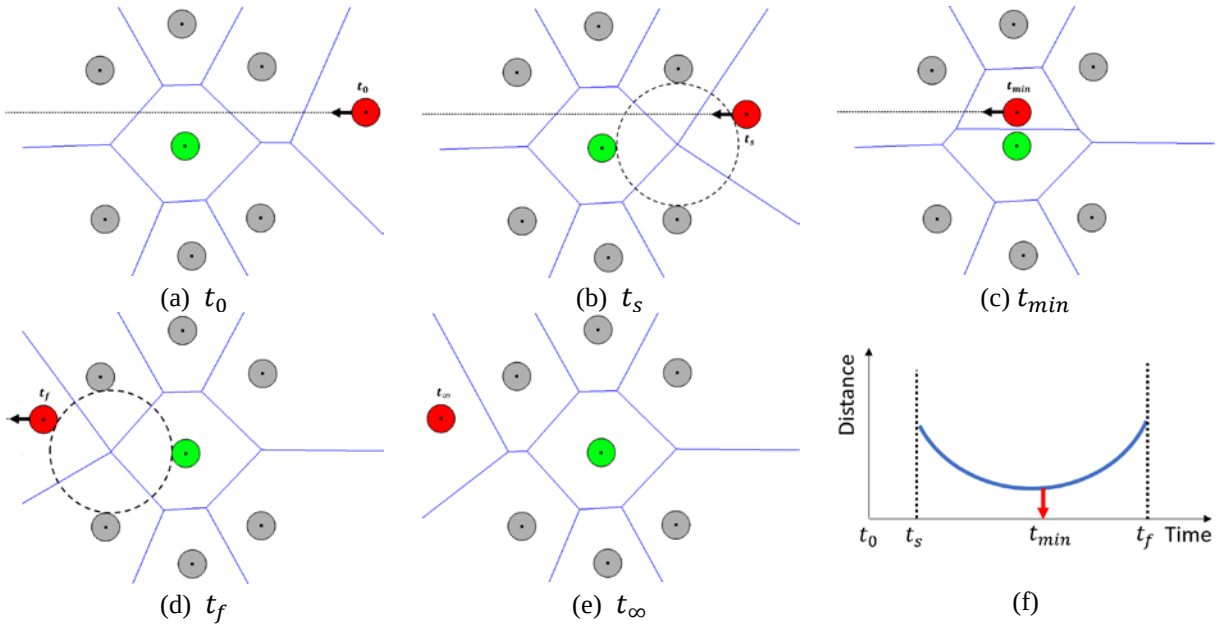
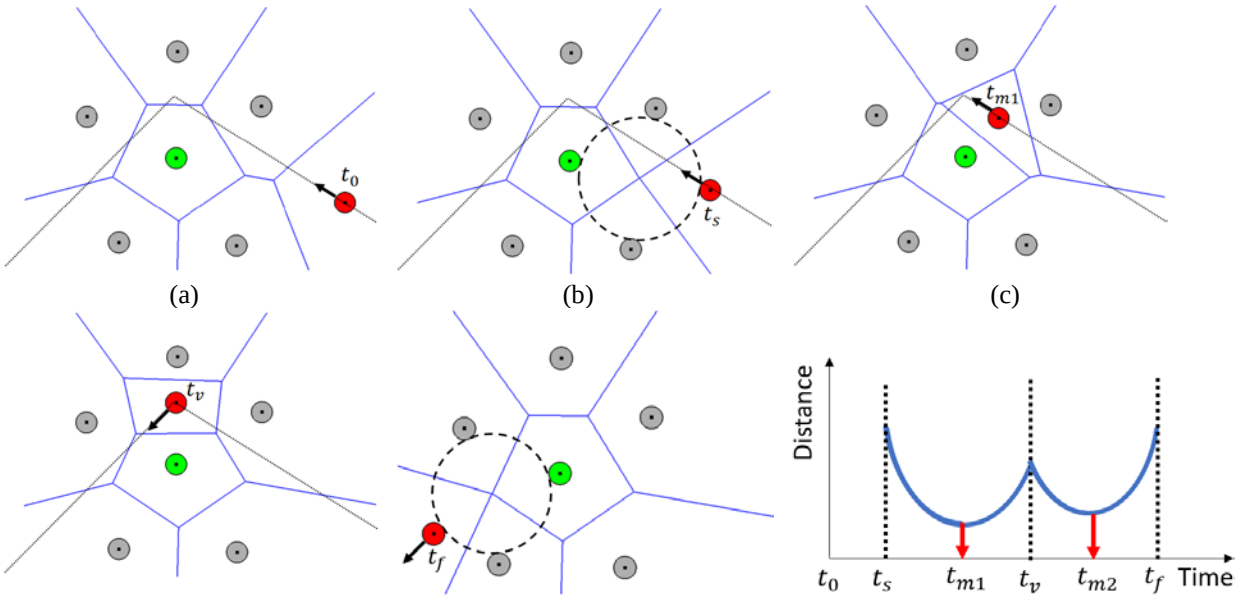


Fig. 1. An example of distance curve generation using the COOP-HSTRY data in the plane without velocity change. Suppose that we want to analyze the distance between the red(moving) and green(fixed). (a) Initial state at time t_0 (b) Edge-flip event time. (c) Moment of local closest approach. (d) Another edge-flip event time. (e) Arrival time. (f) Arbitrary distance curve between the red and green.

The non-zero segment of the distance function is bounded by the two times, say t_s and t_f, given by the two edge-flip events available from the COOP-HSTRY data.

Fig. 2 shows a more general example with a velocity change while two particles maintain their V-neighborhood. In this example, all conditions are identical the previous example of Fig. 1 except the red changes its velocity vector. When the velocity change event occurs at t_v, as shown in Fig. 2(c), the distance function changes as well. Therefore, the whole non-zero distance function of this example consists of two curve segments where each is defined by Eq. (6) and has its own local minima. The following Lemma can be proved without much difficulty based on the observation above.



(a) (b) (c) (d) (e) (f)

Fig. 2. An example of distance curve generation using the COOP-HSTRY data in the plane with velocity change. Suppose that we want to analyze the distance between the red(moving) and green(fixed). (a) Initial state at time t_0 (b) Edge-flip event time. (c) Moment of one local closest approach. (d) velocity change event time. (2) Another edge-flip event time. (f) Arbitrary distance curve between the red and green.

Lemma 4. Suppose that two objects move linearly with each constant speed and the two objects are V-neighbors between t_s and t_f.

Suppose that there are h_v velocity changes in COOP-HSTRY data between t_s and t_f. Then, the distance function consists of $h_v + 1$ non-zero curve segments where the junctions between two consecutive segments are given by the velocity-change events in the COOP-HSTRY data.
In 2D, the distance between two replicas begins to be defined when the V-edge between them is created via an edge-flip operation and vanishes when the V-edge vanishes via another edge-flip operation. In 3D, the distance between two replicas begins to be defined when the V-face between them is created via the edge-flip operation and vanishes when the V-face between them vanishes via a face-flip. Note that, in 3D, a V-face flips to a V-edge and a V-edge flips to a V-face.

Let $|F_{initVD}|$ be the number of the V-faces in the initial Voronoi diagram and $|H_{eFlip}|$ the number of the edge-flip events in the COOP-HSTRY file. Then, in 3D, the number of total distance functions defined within the entire prediction time window is $|F_{initVD}| + |H_{eFlip}|$, which is the number of V-faces. For each distance function, a velocity change creates a new segment of distance function. Thus, the total number of distance function segments within the entire prediction time window is $O(|F_{init}| + |H|)$, where the $|H|$ is the size of COOP-HSTRY during the entire prediction time window. As each distance curve can be computed in a constant time, the next theorem holds.

Theorem 5. Time and space complexities of the CIS-problem are $O(|F_{init}| + |H|)$ in the worst case.

We want to emphasize, in 2D, that the distance function between two RSOs is defined if, and only if, the two RSOs share a Voronoi edge. Hence, there are O(n) distance functions in the system at an arbitrary moment for n RSOs as there are O(n) Voronoi edges. In 3D, there can be $O(n^2)$ distance functions for n RSOs because two RSOs define a distance function if, and only if, the two RSOs share a Voronoi face.

3.2 Maneuver Path Optimization (MPO) Problem

If two RSOs are predicted to be sufficiently close to require a collision mitigating maneuver, we need to maneuver one RSO so that the collision risk can be resolved. Given a configuration of two RSOs, there are usually many alternatives of maneuvers and one would be better than another in terms of a given measure. Hence, these alternatives need to be evaluated by the measure. To the best of our knowledge, subjective judgements by human experts are inputs to the decision-making process of these maneuvers.

In this paper, we propose an efficient, convenient, and robust method using the DVD-COOP algorithm and COOP-HSTRY data as follows. Consider a CI-quadruplet $\{o_i, o_j, t_{start}, t_{end}\}$ with $d_{ij} \leq \theta$ for the threshold θ. We want to maneuver o_i , called an **intruding RSO** or **intruder**, while o_j stays on its original orbit. Suppose that the COOP-HSTRY data and a CIS-problem solver are available. Then, we take the "**generate-and-test approach"** in that we first quickly generate multiple **hypothetical maneuver paths** for o_i and test each hypothesis by solving the CIS-problem quickly. We call the imaginary RSO orbiting through the hypothesized path an **image** of the intruding RSO. We simply collect the hypothesized paths which are guaranteed to be safe and choose the best one in the given measure as the optimal maneuver path. For simplicity in this paper, we generate hypothesized paths by modifying the orbiting plane. However, in practice, a maneuver hypothesis has either a higher or a lower altitude than its original orbit.

In this section, we explain an algorithm to evaluate a hypothesized path using the COOP-HISTRY data. Note that the hypothesized path, also its image on the hypothesized path, does not exist in the COOP-HSTRY data. To solve the CIS-problem for each hypothesized path, it is necessary to find the neighbor of the object image on the hypothesized path to define the inter-object distance. Thus, the MPO-problem solving process consists of two steps for each hypothesized path: i) Search the neighbors and ii) solve the CIS-problem.

3.2.1 Neighbor Search for each Hypothesized Maneuver Path

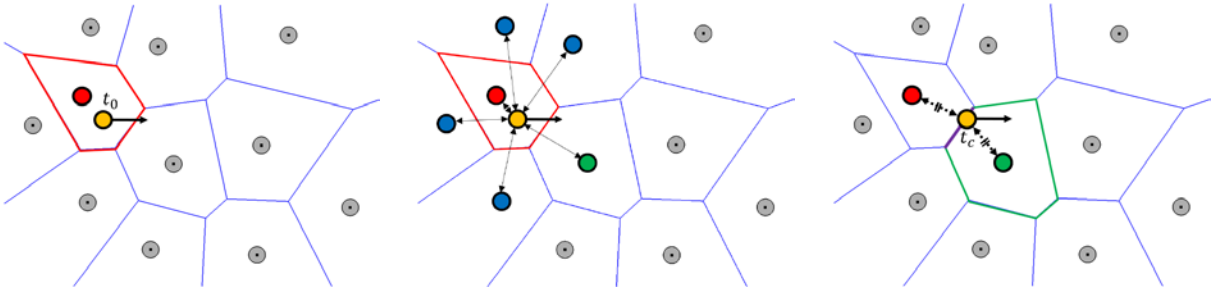
We want to solve the CIS-problem of the imaginary object *I* on the hypothesized path π against all RSOs, but using the replicas of the RSOs. In order to do it, we also need to create the replica of the image *I* following through a piecewise linear path. Let $\hat{I}$ be the replica of *I* and $\hat{o}_{nearest}(t)$ be the nearest neighbor of $\hat{I}$. This means that the nearest neighbor of $\hat{I}$ is the owner of the Voronoi cell that contains $\hat{I}$. The set of neighbors of $\hat{I}$ is its nearest neighbor $\hat{o}_{nearest}(t)$ plus the neighbors of $\hat{o}_{nearest}(t)$ in a similar concept introduced earlier. Note that $\hat{o}_{nearest}(t)$ is indeed the replica of an orbital object. Let d_i^I be the Euclidean distance between *I* of an intruder on a hypothesized path π and a RSO o_i on an arbitrary orbit. Let $\hat{d}_i^I$ be the counterpart of d_i^I for the replicas.

For a given moment *t*, we can find $\hat{o}_{nearest}$ simply by finding the owner of the Voronoi cell containing $\hat{I}$. To find $\hat{o}_{nearest}(t)$ for an arbitrary moment *t*, we trace the COOP-HSTRY data until *t* is reached while we detect the moment, called the **transition moment**, that the nearest neighbor is changed by the motion of $\hat{I}$. We call this process the **Nearest Neighbor Traversal** (NNT). Given an initial nearest neighbor $\hat{o}_{nearest}(t)$ of $\hat{I}$, the moment that the nearest neighbor changes is one that the distance from $\hat{I}$ to $\hat{o}_{nearest}(t)$ is equal to the distance from $\hat{I}$ to one of the neighbors of $\hat{o}_{nearest}(t)$. Therefore, the following lemma is proved.

Lemma 6. Let $\hat{o}_\sigma$ be one of the Voronoi neighbors of $\hat{o}_{nearest}(t)$. Then, a nearest neighbor transition occurs when $\hat{d}_{nearest}^I = \hat{d}_\sigma^I$.

If all transition moments are computed for a given initial nearest neighbor, $\hat{o}_{nearest}(t)$ for arbitrary time t can be found.

(Example: 2D) Fig. 4 shows an example of the NNT process. Orange ball in Fig. 4(a) denotes $\hat{I}$ and red ball denotes the nearest neighbor $\hat{o}_{nearest}(t_0)$, respectively. Fig. 4(b) shows the Voronoi neighbors of $\hat{o}_{nearest}(t_0)$ (i.e. the blue and green balls). Suppose that the green ball has the transition moment t_c. Then, at t_c in Fig. 4(c), the orange ball crosses the Voronoi edge between the Voronoi cells of the red and green balls and the nearest neighbor of the orange changes to the green ball at t_c.



(a) (b) (c)

Fig. 4. An example of the NNT process. (a) The initial state at t_0 with an image (i.e. the orange ball) and its nearest neighbour (i.e. the red ball). (b) Transition moment search with the neighbours of the current nearest neighbour (i.e. the green and blue balls). (c) Transition moment t_c when the orange ball crosses the Voronoi edge between the red and green balls.

The transition moment needs to be updated when the following occurs:

1. The velocity of any of the associated balls (i.e. the red, blue, and green balls) changes.
2. One of the V-neighbors are changed through the topology update.

The computation of the transition depends on the size of COOP-HSTRY and the following lemma can be proved.

Lemma 7. For a given imaginary RSO *I*, the nearest neighbor for the entire prediction time window can be found in $O(|H|)$ in the worst case.

3.2.2 CIS-problem for an Imaginary RSO

The CIS-problem for an imaginary RSO *I* can be solved as before except that the neighbor changes not only when topology update occurs but also when a nearest neighbor transition occurs.

(Example: 2D) Fig. 5 shows an example of the process to track the inter-object distance using the COOP-HSTRY data in the plane. In Fig. 5, suppose that we want to analyze the distance between the replica of the imaginary RSO $\hat{I}$ (i.e. the orange) and the fixed object (i.e. the green) where the orange moves horizontally. The nearest neighbor and its V-cell are shown as the red ball and polygon, respectively. Fig. 5(a) is the initial state at t_0: The distance between the orange and green balls is not defined because the green is not a neighbor of the red. At t_1 , the first transition in Fig. 5(b), the nearest neighbor is changed so that the green becomes a neighbor of the orange. Thus, the distance begins to be defined from t_1. In Fig. 5(c) and (d), new transitions occur at t_2 and t_3, respectively, and the neighbourhood condition between the orange and green is maintained until t_3. After t_3, as shown in Fig. 5(e), they are not neighbor anymore and thus no distance function can be defined. The distance function from t_1 to t_3 consists of one, and only one, curve segment as there is no velocity change at all.

In summary, there are three events in processing the COOP-HSTRY data as follows:

- **Topology update event**: If a topology update changes the neighbor of the candidate, a distance function is either created or disappears.
- **Velocity change event**: If a neighbor replica changes its velocity, a new segment of distance function is created.

- **Nearest neighbor transition event**: Given a transition, a distance function is created or disappears for the new neighbor or the old neighbor, respectively. The distance functions for all the common neighbors for the new nearest and old nearest neighbors remain identical.

The number of distance function segments for an imaginary RSO depends on the size of the COOP-HSTRY data. Thus, the worst-case time complexity of CIS for the hypothesis is $O(|H|)$.

Lemma 8. There are $O(|H|)$ segments of distance functions for an imaginary RSO over the entire prediction time window in the worst case.

Corollary 9. The CIS-problem of an imaginary RSO can be solved in $O(|H|)$ time in the worst case.

Lemma 7 and Corollary 9 lead to the following theorem.

Theorem 10. The optimal maneuver plan among *m* hypothesis can be found in $O(m|H|)$ time in the worst case.

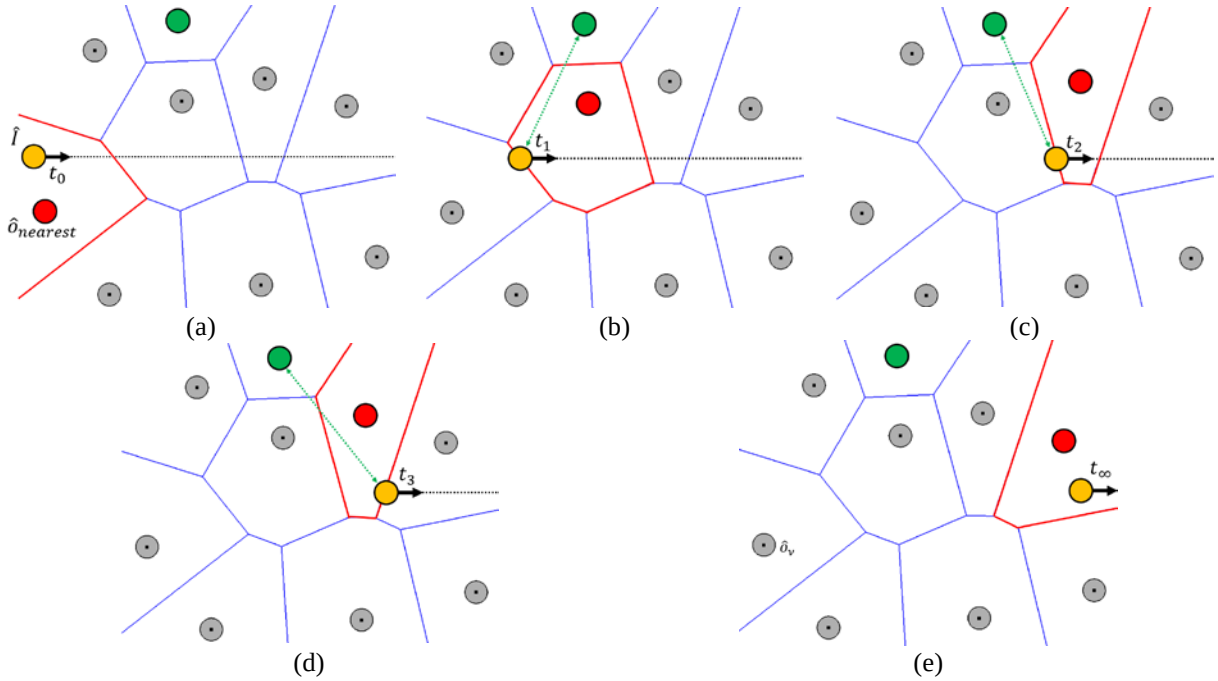


Fig. 5. An example of the distance function generation for the replica of an imaginary RSO using the COOP-HSTRY data with a velocity change in the plane. We want to study the distance between the orange and the green. (a) Initial state at t_0 (b) First transition at t_1: The green becomes a neighbour of the yellow. (c) Next transition. (d) The last transition at t_3: The orange and green are no more neighbours. (e) No distance defined after t_3.

4. Scalability of the Conjunction-prediction using DVD-COOP

The conjunction-prediction using the DVD-COOP algorithm is scalable in many aspects. In addition to the prediction-window scalability and flip-influence scalability as described in [44], there are two more important scalabilities: Neighbor-pair scalability (NP-scalability) and Hypothesis-evaluation scalability (HE-scalability). NP-scalability is based on the observation that the conjunction interval search for each distance function can be done independently. Suppose that we have found the distance functions between the replica pairs. The conjunction interval for replicas can be found instantly from the distance function. To find the true conjunction interval for the orbiting RSOs and the TCA, the solution space search in the conjunction interval for replicas is needed. The solution process of the CIS-problem (distance function → CI for replicas → CI for RSOs → TCA) is independent of each distance function and thus can be solved independently.

HE-scalability is based on the observation that the evaluation of each hypothesized maneuver path can be processed independently. Suppose that we have generated several hypotheses that follow different orbits. Each hypothesis needs to be evaluated in terms of its CI and TCA and the evaluation for each hypothesis can be processed independently. Recall that each hypothesis evaluation can also be parallelized by the NP-scalability. Fig. 6 summarizes the structure of these scalabilities. The current version of DVD-COOP runs on a single core: The computation speed can be significantly improved if these scalabilities are properly incorporated. We plan to implement the scalable version of the proposed algorithm soon.

Scalabilities in DVD-COOP

	COOP-HSTRY Generation	CIS-problem	MPO-problem
Scalability	**Prediction-window scalability(PW)** : Depends on #Cores.		
	Flip-influence scalability(FI) : Depends on adjacent Voronoi entities(≈ 10)	**Neighbor-pair scalability(NP)** : #Neighbor pairs (≈ #V-faces/#Cores)	**Hypothesis-evaluation scalability(HE)** : #Hypothesis

Fig. 6. The structure of scalability properties in the DVD-COOP algorithm.

5. Results

We have implemented the proposed maneuver path optimization algorithm and tested for performance evaluation. Computation environment: i7-6700 3.4Ghz CPU, 16Gb RAM, Microsoft Visual Studio C++ 2017. Note that we used only one core in this experiment. All data used in the experiment was downloaded from JSpOC TLE data.

5.1 Computation Time

Fig. 7(a) depicts the computation time for generating the COOP-HSTRY data vs. the number of RSOs. The curve shows a super linear pattern. Fig. 7(b) depicts the search time for conjunction interval (CI-search time) vs. the number of objects: It shows a super linear pattern. Fig. 7(c) depicts the computation time for generating the COOP-HSTRY data vs. the number of line segments for approximating each orbit: It shows a linear pattern. Fig. 7(d) depicts the CI-search time vs. the number of line segments for approximating each orbit: It shows a linear pattern. Summary: Both the COOP-HSTRY generation time and the CI-search time show super linear trends to the number of RSOs whereas both show linear trends to the number of line segments per orbit. It is known that the COOP-HSTRY generation time is in proportion to the size of the COOP-HSTRY data and thus it can be said that the CI-search time is positively correlated with the size of the COOP-HSTRY data: This observation supports Theorem 2 and 10.

5.2 Maneuver Path Optimization

We have tested the maneuver path optimization for the Korean KOMPSAT satellite series against 1,000 RSOs down loaded on July 10, 2018 from the low earth orbit (LEO) NORAD TLE data. We predicted 10,000 seconds (about 3h) from 0:00 AM of July 12, 2018. In this experiment, we changed the angle of the plane containing the orbit of each satellite to generate hypothesized maneuver path: This hypothesis generation might be rather simple to be realistic but is used in this study to demonstrate the capability of the DVD-COOP for SSA and STM.

The results are shown in Fig. 8. See Fig. 8(c): The green curve represents the orbit of KOMPSAT-4 (NORAD ID 40536) RSO shown as the green ball. Note that the ball radius is 100km. DVD-COOP reported the KOMPSAT-4 was swing by 38km away from the RSO 15505 after 4,742 seconds. See the zoom-up view shown as the box in the right: The green and red balls are very close. The blue curve and ball similarly represent the hypothesized path and the imaginary RSO. In the examples in Fig. 8, except Fig. 8(d) for 10 degrees, we rotate the planes containing the

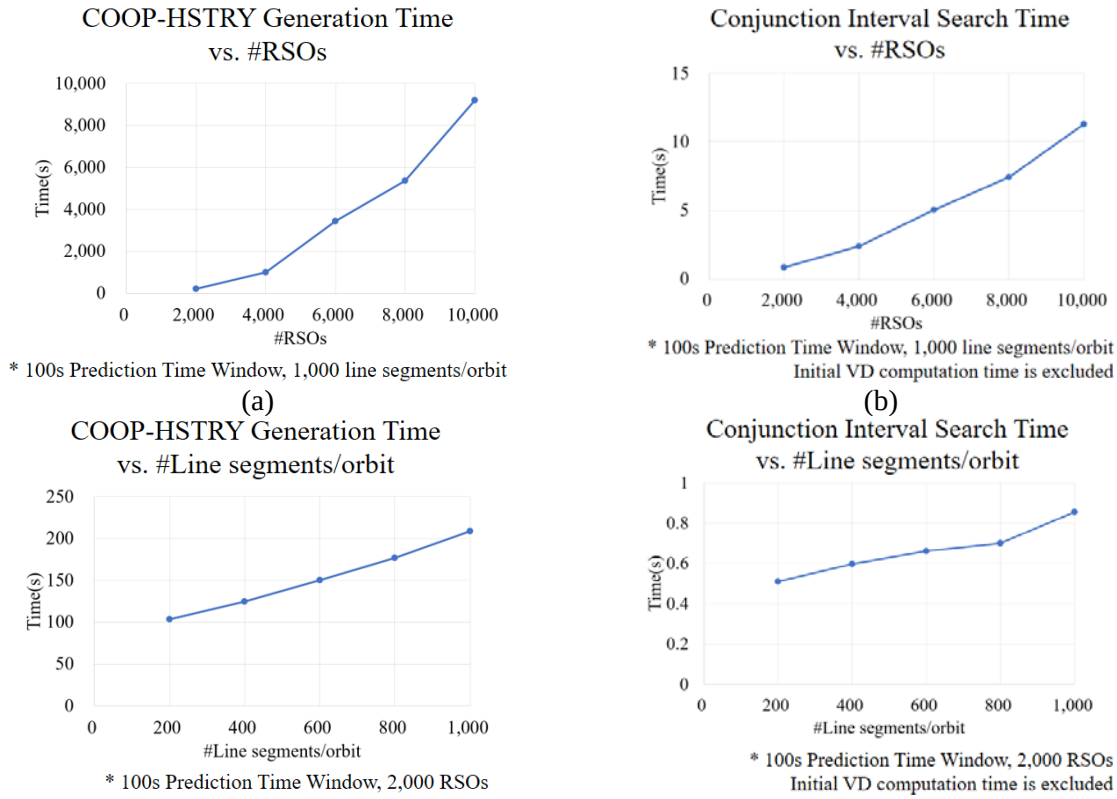


Fig. 7. Computation time graphs. (a) COOP-HSTRY generation time vs. #RSOs. (b) Conjunction Interval Search Time vs. # RSOs (c) COOP-HSTRY generation time vs. #Line segments/orbit. (d) Conjunction Interval Search Time vs. #Line segments/orbit

orbits 20 degrees about the line passing through the RSO position at the initial location at $t = 0$ and the center of the earth. Running the MPO-problem with one hypothesized maneuver path of the given rotation angle, we find the TCA situation predicted by the CIS-problem is resolved. The blue balls represent the imaginary RSOs at the identical time with the green RSOs which were at TCA situations. Recall that we currently solve this problem using replicas: This is the reason why the blue ball in Fig. 8(b) is off the blue curve in the zoom-up. This observation does not preclude the existence of a TCA of the imaginary RSO with another RSO. In this example of Fig. 8(c), the solving the CIS-problem of the hypothesized path against all RSOs yields a new TCA with the distance 76km with RSO 19256 in a certain time within the prediction time window. Hence, we can conclude from this computational experiment that the imaginary RSO following the hypothesized path is safer than the original. The other figures show other KOMPSAT series satellites: KOMPSAT-2 (Fig. 8(a)); KOMPSAT-3 (Fig. 8(b)), and KOPSAT-5 (Fig. 8(d)).

6. Conclusion

Collisions are critical conditions for Space Situational Awareness and Space Traffic Management and thus their detection and resolution are critical for safe and efficient utilization of geospacer. In this paper, we proposed the DVD-COOP algorithm, based on the dynamic Voronoi diagram of 3D spherical balls, which is general-purpose, accurate, efficient for the conjunction predictions and resolutions. This proposal is based on our earlier work reported in AMOS2017.

We report two algorithms with the DVD-COOP framework: Conjunction-interval search (CIS) problem and maneuver path optimization (MPO) problem. The CIS-problem quickly finds all conjunction intervals with a guarantee of no-missing solution and the MPO-problem finds the optimal maneuvering orbit by solving the CIS-problem for sufficiently many hypothetical paths.

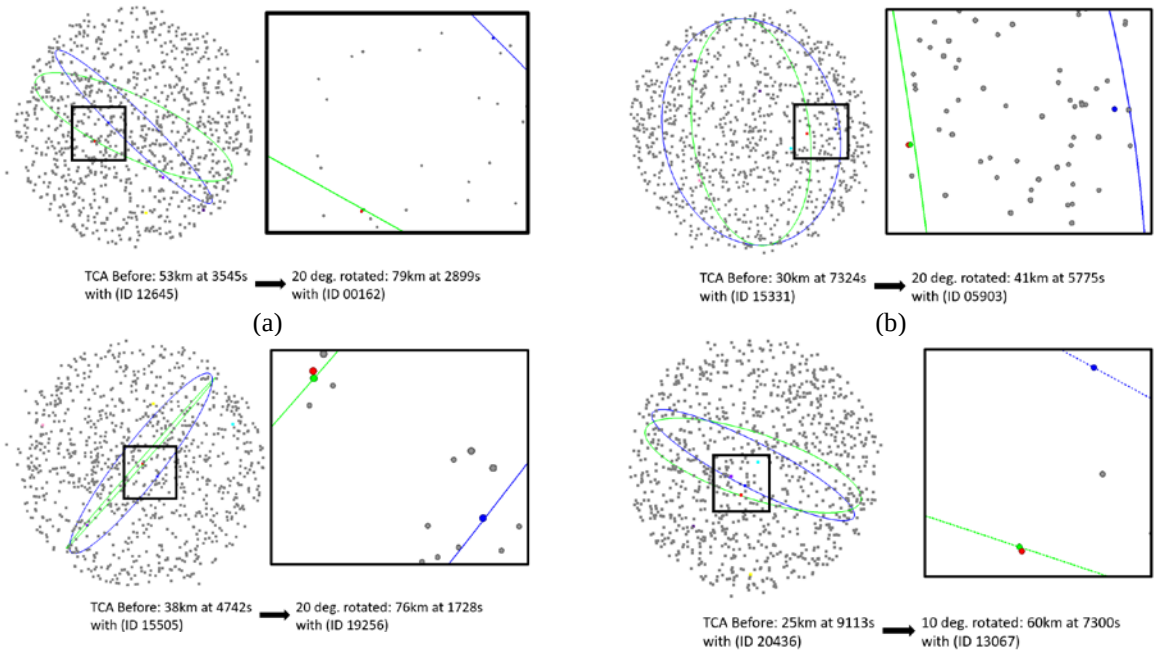


Fig. 8. MPO-problems for four satellites in the KOMPSAT series. The green curve is the orbit of the satellite shown as the green ball. The red ball represents the secondary RSO at the TCA. The blue curve represents a hypothesized path of an imaginary RSO shown as the blue ball. (a) KOMPSAT-2 (NORAD ID 29268). (b) KOMPSAT-3 (NORAD ID 38338). (c) KOMPSAT-4 (NORAD ID 40536). (d) KOMPSAT-5 (NORAD ID 39227).

The proposed algorithm was implemented in Microsoft Visual Studio C++ 2017. Even if there are several scalabilities in the proposed algorithm, the current implementation uses only one core: Hence, the implementation with scalability is a challenge we are working on now. We believe that the scalable implementation of the DVD-COOP algorithm will play a key role for Space Situational Awareness and Space Traffic Management and we hope it soon used by researchers in this community.

7. Acknowledgement

This work was in part supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP, MSIT) (No. 2017R1A3B1023591) and in part by the Asian Office of Aerospace R&D of USA (No. FA2386-17-1-4050)

8. 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. Radtke, J. et al. The impact of the increase in small satellite launch traffic on the long-term evolution of the space debris environment, In Proceedings of the 7th European Conference on Space Debris, 2017.
5. Liou, J.C. and Johnson, N.L., Risks in Space from Orbiting Debris, Science, Vol. 311, 340-341, 2006.
6. Peeters, W. From suborbital space tourism to commercial personal spaceflight. Acta Astronautica, 66(11-12), 1625-1632, 2010.
7. Chang, Y. W. The first decade of commercial space tourism. Acta Astronautica, 108, 79-91, 2015.
8. Johnson, N. L. Space traffic management concepts and practices. Acta Astronautica, 55(3-9), 803-809, 2004.
9. Cukurtepe, H. and Akgun, I. Towards space traffic management system. Acta Astronautica, 65(5-6), 870-878, 2009
10. Budianto-Ho, I.A. et al, Scalable Conjunction Processing using Spatiotemporally Indexed Ephemeris Data, In Advanced Maui Optical and Space Surveillance Technologies Conference, 2014.
11. Alarcón-Rodríguez, J.R. et al, Conjunction event predictions for operational ESA satellites, In Proceedings of the Space-Ops 2004 Conference, 2004.
12. Alarcón-Rodrıguez, J.R. et al, Development of a collision risk assessment tool, Advances in Space Research, Vol. 34, 1120-1124, 2004.
13. 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.
14. Wisniowski, T. and Rickman, H., Fast Geometric Method for Calculating Accurate Minimum Orbit Intersection Distances, Acta Astronomica, Vol. 63, 293-307, 2013.
15. Chen, H. et al, Conformal Prediction for Anomaly Detection and Collision Alert in Space Surveillance, In Proceedings of SPIE 8739, 2013.
16. Alfano, S., Toroidal Path Filter for Orbital Conjunction Screening, Celestial Mechanics and Dynamical Astronomy, Vol. 113, 321-334, 2012.
17. Crassidis, J. et al, Space Collision Avoidance, In Proceedings of National Symposium on Sensor and Data Fusion, 2011.
18. 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.
19. 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.
20. Healy, L.M., Close Conjunction Detection on Parallel Computer, Journal of Guidance, Control, and Dynamics, Vol. 18, 824-829, 1995.
21. 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.
22. Carlson, R. and Lee, J., Detecting Near Collisions for Satellites, IEEE transactions on aerospace and electronic systems, Vol. 33, 921-929, 1997.
23. Hoots, F.R. et al, An Analytic Method to Determine Future Close Approaches between Satellites, Celestial mechanics, Vol. 33, 143-158, 1984.
24. Kubica, J. et al, Efficient Intra-and Inter-night Linking of Asteroid Detections Using Kd-trees, Icarus, Vol. 189, 151-168, 2007.
25. Kouprianov, V., ISON Data Acquisition and Analysis Software. In 6th European Conference on Space Debris, 2013.
26. Denneau, L. et al, The Pan-STARRS Moving Object Processing System. Publications of the Astronomical Society of the Pacific, Vol. 125, 357, 2013
27. Mercurio, M. et al, A Hierarchical Tree Code Based Approach For Efficient Conjunction Analysis, In AIAA/AAS Astrodynamics Specialist Conference, 2012.
28. Bentley, J.L., Multidimensional Binary Search Trees Used for Associative Searching., Communications of the ACM, Vol. 18, 509-517, 1975.
29. Alarcón-Rodriguez, J.R. et al, Collision Risk Assessment With a "Smart Sieve" Method, In Joint ESA-NASA Space-Flight Safety Conference, 2002.

30. Okabe, A. et al, Spatial Tessellations - Concepts and Applications of Voronoi Diagram, John Wiley & Sons, 1992.
31. 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.
32. 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.
33. 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.
34. Kim, D. and Kim, D.-S., Region-Expansion for the Voronoi Diagram of 3D Spheres, Computer-Aided Design, Vol. 38, 417-430, 2006.
35. Kim, D.-S. et al, Quasi-Triangulation and Interworld Data Structure in Three Dimensions, Computer-Aided Design, Vol. 38, 808-819, 2006.
36. Kim, D.-S. et al, Three-Dimensional Beta Shapes, Computer-Aided Design, Vol. 38, 1179-1191, 2006.
37. Kim, D.-S. et al, Quasi-worlds and Quasi-operators on Quasi-triangulations, Computer-Aided Design, Vol. 42, 874-888, 2010.
38. Kim, D.-S. et al, Three-dimensional Beta-shapes and Beta-complexes via Quasi-triangulation, Computer-Aided Design, Vol. 42, 911-929, 2010.
39. Kim, D.-S. et al, Querying Simplexes in Quasi-triangulation, Computer-Aided Design, Vol. 44, 85-98, 2012.
40. Kim, D.-S. et al, QTF: Quasi-triangulation File Format, Computer-Aided Design, Vol. 44, 835-845, 2012.
41. 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.
42. 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.
43. 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.
44. Cha, J. 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.