

Article

Performance Investigation of the Conjunction Filter Methods and Enhancement of Computation Speed on Conjunction Assessment Analysis with CUDA Techniques

Phasawee Saingyen ^{1,2}, Sittiporn Channumsin ^{2,*}, Suwat Sreesawet ², Keerati Puttasuwan ^{1,2} and Thanathip Limna ^{1,*}

¹ Department of Computer Engineering, Faculty of Engineering, Prince of Songkla University, Songkhla 90110, Thailand; phasawee@gistda.or.th (P.S.); keerati.p@gistda.or.th (K.P.)

² Space Technology Research Center, Geo-Informatics and Space Technology Development Agency (GISTDA), Chonburi 20230, Thailand; suwat@gistda.or.th

* Correspondence: sittiporn@gistda.or.th (S.C.); thanathip.l@coe.psu.ac.th (T.L.)

Abstract: The growing number of space objects leads to increases in the potential risks of damage to satellites and generates space debris after colliding. Conjunction assessment analysis is the one of keys to evaluating the collision risk of satellites and satellite operators require the analyzed results as fast as possible to decide and execute collision maneuver planning. However, the computation time to analyze the potential risk of all satellites is proportional to the number of space objects. The conjunction filters and parallel computing techniques can shorten the computation cost of conjunction analysis to provide the analyzed results. Therefore, this paper shows the investigation of the conjunction filter performances (accuracy and computation speed): Smart Sieve, CSieve and CAOS-D (combination of both Smart Sieve and CSieve) in both a single satellite (one vs. all) and all space objects (all vs. all) cases. Then, all the screening filters are developed to implement an algorithm that executes General-purpose computing on graphics processing units (GPGPU) by using NVIDIA's Compute Unified Device Architecture (CUDA). The analyzed results show the comparison results of the accuracy of conjunction screening analysis and computation times of each filter when implemented with the parallel computation techniques.

Keywords: space debris; conjunction assessment analysis; conjunction filter; GPGPU; CUDA



Citation: Saingyen, P.; Channumsin, S.; Sreesawet, S.; Puttasuwan, K.; Limna, T. Performance Investigation of the Conjunction Filter Methods and Enhancement of Computation Speed on Conjunction Assessment Analysis with CUDA Techniques.

Aerospace **2023**, *10*, 543.

<https://doi.org/10.3390/aerospace10060543>

Academic Editor: Vladimir S. Aslanov

Received: 29 March 2023

Revised: 12 May 2023

Accepted: 28 May 2023

Published: 7 June 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In 1957, the successful launch of the first satellite was the beginning of a new era in space exploration. Since then, the number of space debris orbiting the Earth continuously increased which has resulted in a higher potential risk to all active satellites as shown in Figure 1 [1]. In 2009, Iridium-33 had an accident with Cosmos-2251 resulting in the mission ending and generating a huge number of space objects [2]. When the number of space objects increases, at some critical point it could cause a collision cascade; the scenario is well known as “Kessler Syndrome” [3]. Therefore, the mitigation of the number of space objects is important for space sustainability. The conjunction assessment analysis is one of the essential tools for all operators to minimize the collision threat for their own satellites.

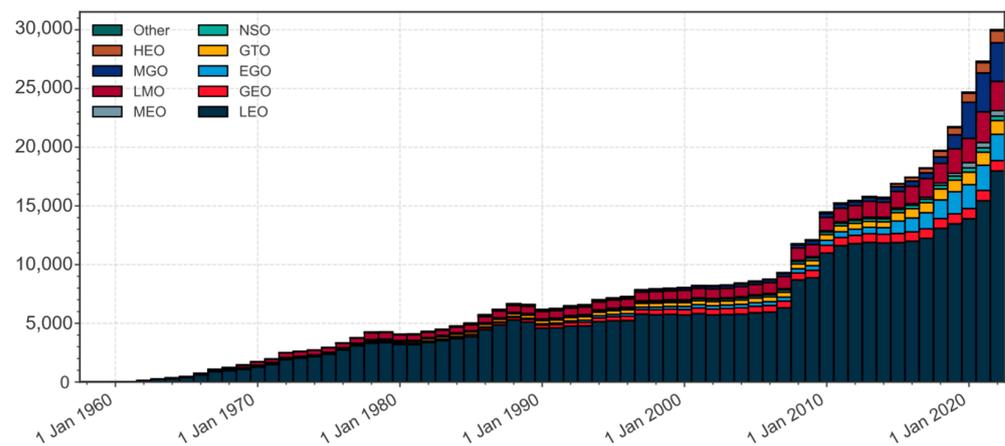


Figure 1. Amount of space objects classified by orbital region (Space environment report by European Space Agency) [4].

The conjunction assessment analysis is a technique to evaluate the risks of collision by defining the conjunctions of each space object pair that exceed a threshold of miss distance, the time of closest approach (TCA) and collision probability. The processing time for conjunction analysis is directly proportional to the number of space objects. Therefore, the efficiency of conjunction assessment analysis is essential for all operations to make critical decisions and execute necessary actions to ensure satellite safety. The conjunction analysis for all space objects (all vs. all), which analyzes all potential risks of all space objects, consumes a significant amount of computation time. To address the challenge of conjunction assessment analysis, many researchers attempt to develop conjunction filter techniques to maintain screening accuracy and enhance the computation cost of collision assessment analysis. Currently, there are three developed screening filter techniques. Firstly, the Smart Sieve method [5] was developed by the European Space Agency (ESA) and implemented in the ESA's conjunction prediction services. Secondly, the CSieve method [6] was developed by the Aerospace Corporation and based on geometrical principles. Lastly, the CAOS-D [7] is the combination of the CSieve and Smart Sieve filters. However, these screening filters require high computation and can miss some conjunction events. Therefore, the first objective of this paper aims to investigate the accuracy of each screening filter to ensure that we can detect all possible conjunction events by comparing them with the conjunction assessment analysis without the screening filters.

To enhance the computation speed of the conjunction assessment analysis with parallel computation techniques, CAOS-D successfully implemented parallel computation on the CPU [7]. Then, ESA implements the Smart Sieve filter on a GPU by using NVIDIA's Compute Unified Device Architecture (CUDA) [8]. The combination can reduce the processing time by 89.3 percent compared to parallel computing on a CPU (four cores). Lin investigated the computing speed on the parallel computation between CPU and GPU, and both of them do not consider the screening filters. The investigated results demonstrate that the computing speed of the parallel algorithm on a GPU is 30 times faster than on a CPU [9]. Furthermore, recent studies implemented deep learning approaches to reduce the computation time of conjunction screening in the case of all vs. all [10]. The investigated results present that a deep learning technique can reduce the computation time while maintaining the accuracy of the conjunction screening. Therefore, the second purpose of this paper is to develop a combination between a screening filter and parallel computation based on the GPU of each filter to enhance computation speed.

This paper proposes to investigate the accuracy performance and computation speed of a combination between screening filter and parallel computation techniques based on a GPU for conjunction assessment analysis of a single satellite (one vs. all) and all space objects (all vs. all). Section 2 describes the methodology of conjunction assessment analysis with a conjunction filter. The experiment methodology and the investigated results of each

filter efficiency and computation speed are presented in Section 3. Then, the discussion is described in Section 4. Finally, Section 5 summarizes the conclusions of this paper.

2. Conjunction Assessment Analysis

This section describes the process of conjunction assessment analysis as shown in Figure 2. The analyzed results (TCA, miss distance, and collision probability) are necessary for evaluating collision risks. The conjunction assessment process consists of four steps: space object propagation, conjunction filter, the closest approach determination and collision probability calculation.

The process requires the initial states of space objects to propagate into forward time in the prediction period. The process of conjunction assessment analysis is shown in Figure 2.

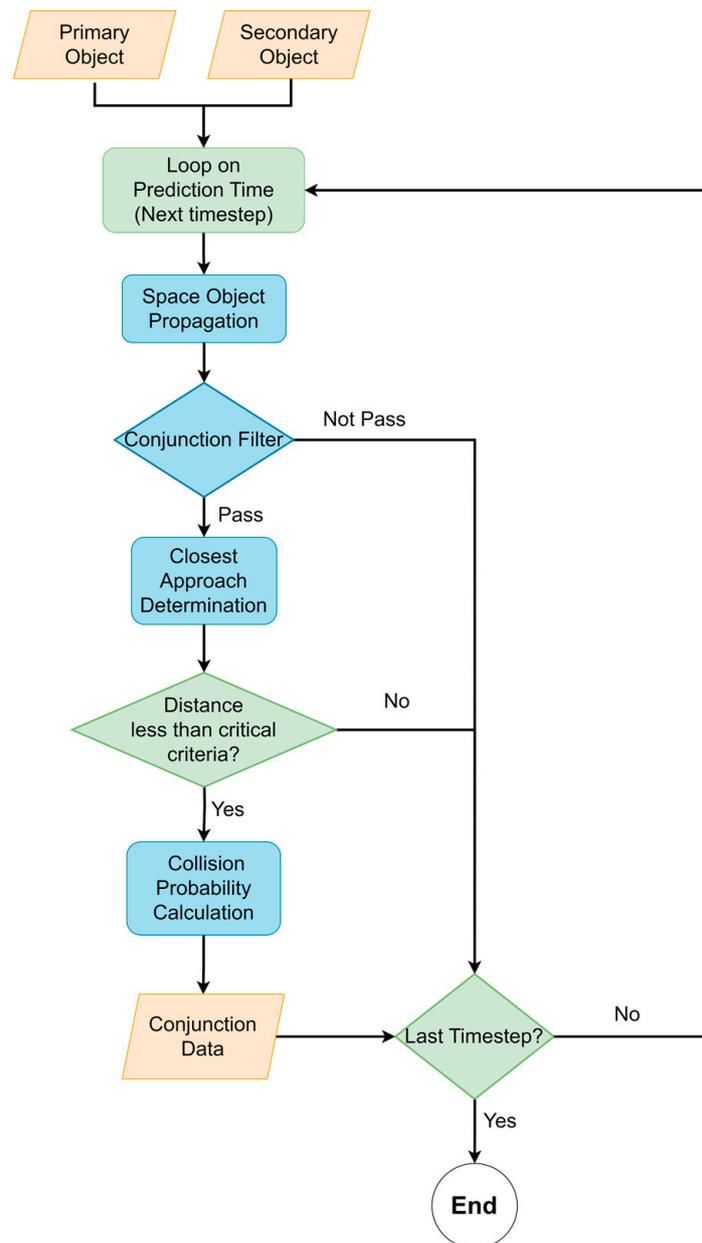


Figure 2. Process of conjunction assessment to analyze conjunction of single pair.

The initial state of both primary and secondary objects in this study is based on the Two-line element (TLE) provided by the space-track [11]. Then, the algorithm performs a loop by timestep from the beginning to the end of the analyzed period. For example,

in the case of one day (1440 min) conjunction analysis with a 2 min timestep, the process looping is $1440/2 = 720$ times. The state vectors (position and velocity) of both objects are propagated into the current loop time and define the time interval that both objects possibly occur in conjunction with events. The time interval that passes the condition of the filter will determine closest approach. The algorithm is designed to calculate a collision probability when a miss distance is less than the threshold control (5 km for this study). The distance between two objects is analyzed in whole time intervals to define the closest approach and calculate collision probability.

For analyzing conjunction events of a single satellite versus all space objects (one vs. all), the total pair (N) between a satellite and all space objects (n) is calculated by:

$$N = n - 1 \quad (1)$$

In the case of all space objects versus all space objects (all vs. all), the total pairing (N) between all space objects can be calculated by:

$$N = \frac{n(n-1)}{2} \quad (2)$$

2.1. Conjunction Filter

Figure 3 shows the concept of conjunction filter. The filter is useful to screen the possible time interval that both objects are not closer within a critical distance (D_{cr}). This supports that the process defines TCA points in some time intervals on the prediction period, which can reduce the computation time. The filter defines the threshold distance (D_{th}) to screen relative distance (r_{rel}) and considers the time interval that the relative distance is within the threshold distance. The relative distance can be calculated by

$$r_{rel} = \left| \vec{r}_2 - \vec{r}_1 \right| \quad (3)$$

where \vec{r}_1 and \vec{r}_2 are the predicted position vectors of both in the current loop time. There are three filter methods (smart sieve, CSieve and CAOS-D) to define all possible times interval to determine conjunction events. The three filter methods are described in sub-sessions.

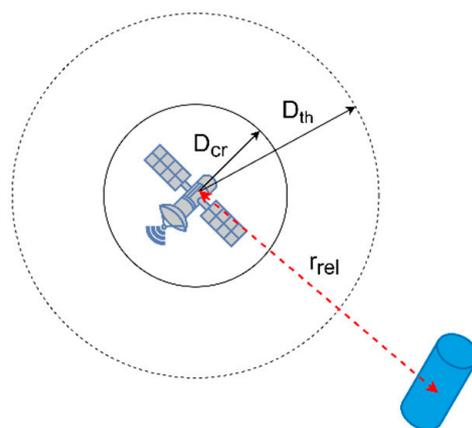


Figure 3. Both Object Positions in Current Loop Time.

2.1.1. Smart Sieve

In the classical sieve [12] method proposes to minimize distance estimation. The remaining time interval and the minimum distance between both space objects are estimated.

The estimated distance will be verified with critical distance (D_{cr}). The minimum distance can be calculated by:

$$r_{\min} = \sqrt{r_{\text{rel}}^2 - \left(\vec{r}_{\text{rel}} \cdot \frac{\vec{v}_{\text{rel}}}{|v_{\text{rel}}|} \right)^2} \quad (4)$$

The Smart Sieve [5] is improved from the classical sieve by using the simplified dynamic model included in the method. The filter defines threshold and acceleration safety distance based on the flight dynamics principle. The threshold distance (D_{th}) is defined by:

$$D_{\text{th}} = D_{\text{cr}} + V_{\text{esc}}\Delta t \quad (5)$$

where V_{esc} is Earth's escape velocity, Δt is the timestep size of loop time. The acceleration safety distance is defined by:

$$D_{\text{acc}} = D_{\text{cr}} + g_0\Delta t^2 \quad (6)$$

where g_0 is the Earth's gravity field. Smart Sieve consists of five screening conditions. The first condition is the single-axis screening. If the difference of any axis is more than D_{th} , the relative distance is not within D_{th} . The single-axis screening is:

$$r_{i2} - r_{i1} > D_{\text{th}} \text{ where } i = X, Y, Z \quad (7)$$

Secondly, R^2 screening eliminates objects that are not within the threshold distance that is calculated by Equation (5). R^2 screening is as follows:

$$r_{\text{rel}}^2 > D_{\text{th}}^2 \quad (8)$$

Thirdly, Minimum Distance screening rejects object pairs that are the estimated minimum distance by Equation (4) and not within the acceleration safety distance calculated by Equation (6). Minimum Distance screening is:

$$r_{\min}^2 > D_{\text{acc}}^2 \quad (9)$$

Fourthly, Fine R^2 screening is similar to R^2 screening but the algorithm requires the approach velocity (V_{app}) instead of escape velocity (V_{esc}). The remaining pair may be closer than the threshold distance, but it is still too far from the critical distance. The screening condition is described as follows:

$$r_{\text{rel}}^2 > \left(D_{\text{acc}} + \frac{1}{2}V_{\text{app}}\Delta t \right)^2 \quad (10)$$

where approach velocity (V_{app}) is defined by

$$V_{\text{app}} = \left| \vec{r}_{\text{rel}} \cdot \frac{\vec{v}_{\text{rel}}}{v_{\text{rel}}} \right| \quad (11)$$

Lastly, Fine Detection screening uses estimates of the closest approach distance by the time approach (t_{app}). The closest approach distance that is not within the critical range is rejected. Fine Detection screening is as follows:

$$r_{\text{rel}} + \left| \vec{v}_{\text{rel}} \right| t_{\text{app}} > D_{\text{cr}} \quad (12)$$

where time approach (t_{app}) is calculated by

$$t_{\text{app}} = - \left(\frac{\vec{r}_{\text{rel}} \cdot \vec{v}_{\text{rel}}}{\vec{v}_{\text{rel}} \cdot \vec{v}_{\text{rel}}} \right) \quad (13)$$

2.1.2. CSieve

The filtering principle of CSieve [6] implements the states of both objects at the beginning and end of a time interval to define the rectangular volume surrounding the path. The overlapping of both rectangular volumes indicates that their objects are approaching each other, as shown in Figure 4.

The filter defines the maximum and minimum values of each axis of the space object position over a time interval. The principle is that the second object's minimum value is not within the range limitation of the primary object's maximum value. The condition to reject the rectangular volume is when the rectangular volume does not overlap. The X, Y and Z-axis screening conditions are as follows:

$$r_{i2,min} > r_{i1,max} + D_{cr} \text{ where } i = X, Y, Z \tag{14}$$

Then, the dot product of relative position and velocity vectors at the beginning and end of the time interval is calculated. The condition accepts the time interval that the dot product of the relative position and velocity vectors change from negative to positive during the time interval. The R dot V screening condition definition is as follows:

$$\vec{r}_{t0} \cdot \vec{v}_{t0} < 0 \text{ and } \vec{r}_{t1} \cdot \vec{v}_{t1} > 0 \tag{15}$$

The point of minimum distance corresponds to the dot product equal to zero as follows:

$$\vec{r}_{t0} \cdot \vec{v}_{t0} = 0 \tag{16}$$

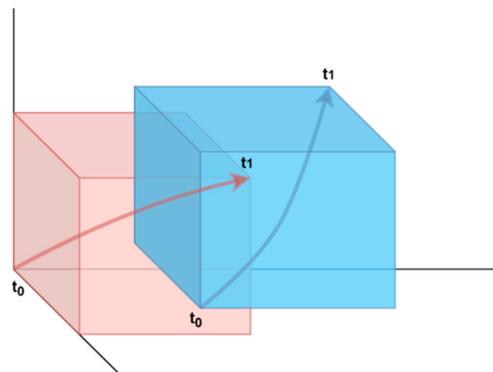


Figure 4. Overlapping of Rectangular Volumes.

2.1.3. CAOS-D

CAOS-D [7] is conjunction assessment analysis software that has been designed to swap differential parts in the algorithm such as orbit propagation, conjunction filter, closest approach determination, etc. The software algorithm combines Smart Sieve and CSieve filter methods to screen for each timestep. The software tests both filters to define the best filter chain. The filter chain that had the best performance in CAOS-D testing is mapped in Figure 5.

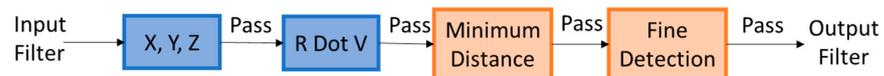


Figure 5. The CAOS-D filter chain combines CSieve (blue) and Smart Sieve (orange) screening condition sets.

The X, Y and Z-axis filter of CSieve is the first stage of the screening. Next, the R dot V guarantees to define conjunction events in the current time interval, but it does not guarantee that the close approach is within critical range. The Minimum Distance and Fine

Detection of the Smart Sieve can greatly screen the number of pairs that pass to the closest approach determination.

2.2. Closest Approach Determination

Figure 6 shows a concept of the closest approach determination. The time interval, which passes the filter, is determined closest approach. This process is to define a point that has a minimum distance of both objects by propagating with a timestep in the second period of the time interval. State vectors are predicted from the beginning (t_s) to end (t_E) of the time interval of interest with the timestep in seconds. The relative distance (r_{rel}) is calculated to determine the time of the closest approach (t_{tca}). If the minimum distance is less than the critical distance (D_{cr}), a conjunction event is detected. The algorithm implements the closest approach determination by propagating with the timestep in the second, to search for a point that the dot product of the relative position and velocity closest to zero. In terms of the timestep in milliseconds, the time of the closest approach is estimated by the Healy equation [12]. The equation is valid for small timestep sizes. The time of the closest approach can be calculated by:

$$t_{tca} = t_0 - \left(\frac{\vec{r}_{rel} \cdot \vec{v}_{rel}}{\vec{v}_{rel} \cdot \vec{v}_{rel}} \right) \tag{17}$$

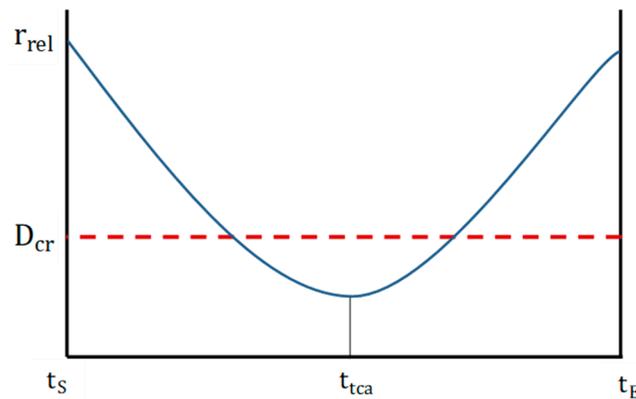


Figure 6. Time Interval of Interest to Determine Closest Approach.

2.3. Collision Probability Calculation

The collision probability (P_c) is calculated by state vectors at the time of the closest approach to evaluate the potential collision. We implement collision probability calculation by following maximum probability [13]. Both objects are assumed to move along straight lines, and position uncertainty during the encounter is assumed to be constant. In this algorithm, the collision probability calculation is implemented the two-dimensional probability density function (PDF) [14]. The collision probability (P_c) can be calculated by:

$$P_c = \frac{1}{2\pi\sigma_x\sigma_y} \int_{-r}^r \int_{-\sqrt{r^2-x^2}}^{\sqrt{r^2-x^2}} \exp^{-\frac{1}{2} \left[\left(\frac{x-xm}{\sigma_x} \right)^2 + \left(\frac{y-ym}{\sigma_y} \right)^2 \right]} dydx \tag{18}$$

where r is the combined object radius, x lies along the minor axis of the error covariance ellipse, y lies along the major axis of the ellipse, σ_x and σ_y are the corresponding standard deviations, and xm and ym are the respective coordinates of the projected distance.

3. Conjunction Filter Investigations

Three conjunction filters are tested to investigate to filter all possible events. The number of conjunction events of each technique and process times are compared with the analyzed results without a conjunction filter. The experiment performs one vs. all and

all vs. all conjunction assessments and initial states of all space objects provided by the space track website. The simplified General Perturbations 4 (SGP4) [15] model provides the propagation error within 1 km on propagation for nine days [16], which is sufficient for this study.

Before testing the filter, we validate our algorithm by comparing the results with Satellite Orbital Conjunction Reports Assessing Threatening Encounters in Space (SOCEATES) [17], which is a service for satellite operators developed by the Center for Space and Innovation (CSSI). The SOCEATES service provides TLEs that are used to compute the conjunction data of their results. For validation, our process is run by using the same TLE data and then comparing the analyzed results (TCA and distance). The maximum difference between miss distance and TCA is 0.015 km and 0.011 s, respectively. The compared results with SOCRATES are shown in Appendix A, Table A1.

3.1. One vs. All Conjunction Assessments

This study selects five satellites in Low Earth Orbit (LEO), Medium Earth Orbit (MEO), and highly elliptical orbit (HEO) to test with three filters. The orbital characteristics of satellites are listed in Table 1.

Table 1. Five satellites in LEO, MEO and HEO orbit to analyze conjunction assessments.

Satellite Name (NORAD)	Period (Minute)	Inclination (Degree)	Apogee (Km)	Perigee (Km)
THEOS (33396)	101.41	98.75	826	824
NAPA-1 (46320)	95.21	97.45	530	528
NAPA-2 (48963)	95.17	97.54	536	518
COSMOS 1675 (15952)	458.42	65.38	25,985	647
IMAGE (26113)	853.83	94.93	45,531	1339

The process without conjunction filter is tested by analyzing the conjunction of satellites in advance seven days, beginning on 19 November 2021 at 00:00:00 UTC and setting critical criteria at 5 km. Space objects (22,119 objects) are loaded from the space track to be the initial states. The process runs on Intel® Core™ i7-8700, which is executed by a single core. The results of the non-filter process of all satellites are shown in Table 2.

Table 2. The analyzed results of screening process without filter.

Satellite Name	Conjunction Event	Computation Time (s)
THEOS	21	32,276.379
NAPA-1	13	32,591.642
NAPA-2	9	32,933.844
COSMOS 1675	4	41,400.025
IMAGE	1	40,974.512

In the case of the non-filter process, all time intervals are analyzed to guarantee all conjunction events that are defined. The analyzed results of each filter are compared to the analyzed results without a non-filter process to investigate the accuracy of each filter.

The filter process run with a timestep of one to ten minutes. Increasing the timestep lead to decrease the time interval to compute the conjunction filter. The experiment purposes to increases a timestep for observing the change in computation time and number of conjunction events. The conjunction events and computation times of five satellites on a process with three filter methods is recorded.

Several time intervals that pass the filter (S_{pass}) are also recorded to verify the screening ratio. The total time interval of all pairs screened (S_{total}) is calculated by loop time size to compute single pairs depending on the prediction period and timestep (Δt).

$$S_{\text{total}} = N \left(\frac{\text{Prediction Period}}{\Delta t} \right) \quad (19)$$

Then, the loop time size multiplied by the total pair (N) is the total screening (S_{total}) of conjunction filter. When increasing the timestep (Δt) length, total screening (S_{total}) is reduced, which minimizes the loop time size for computing.

3.2. All vs. All Conjunction Assessment

It has a considerable object pair that requires high computation process all vs. all conjunction assessment on a single core of CPU. Therefore, we implement parallel computing with a GPU to distribute the process into smaller parts for parallel computation. The process starts by loading all space object data. GPU memory is allocated to send data to propagation. The idea to parallel propagate all space objects onto the prediction period is to compute the object's positions on a timestep per core processing unit. The computation complexity of propagation is $O(n \times s)$ where n is the number of objects and s is the number of timesteps on the prediction period. Then, the process defines object pairs by the reference index of an object into an array list and provides the object pairs to the GPU for screening with a conjunction filter.

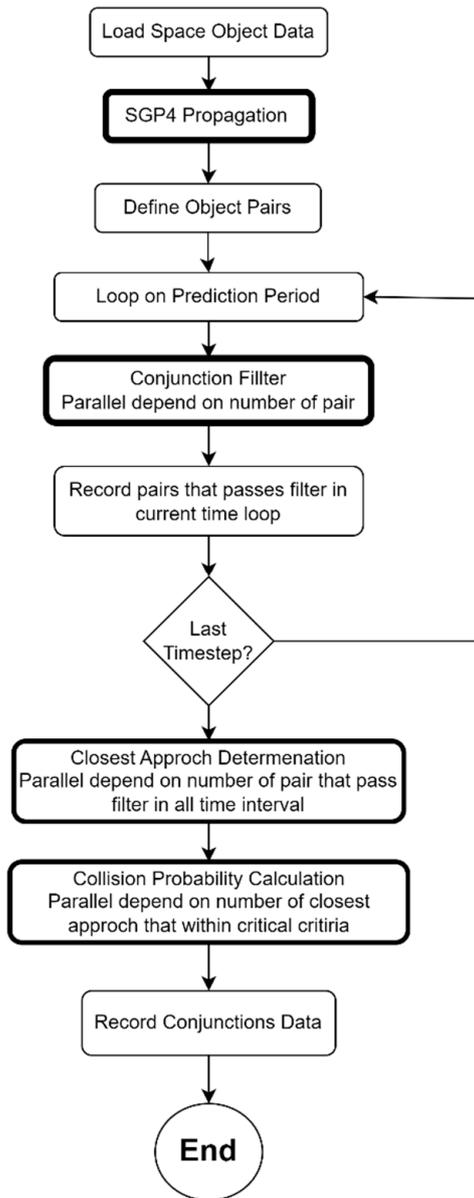
For the parallel conjunction filters, we cannot allocate memory for the result data because it is difficult to define the number of pairs that will pass the filter. However, the process can define the exact number of object pairs that compute on a time interval. This part should be implemented in the critical section or use atomic add but there are significant slowdowns in the runtime process. In this part, we implement a conjunction filter on loop time and parallel computation depends on a number of object pairs. The length of the data is allocated according to the number of pairs. The results of the conjunction filter indicate whether those pairs are passed or not (true or false). Then, the index of object pairs that pass in the current loop time will be recorded to the passed pair list.

The CSieve method cannot directly run on the GPU algorithm because there is a concern of insufficient memory in part of the closest approach determination. Therefore, we develop the algorithm by moving the closest approach determination into the time loop. The CSieve method has a very high screening ratio than other methods. This process will be described in Section 4. The closest approach determination of the modified algorithm is reduced object pairs that compute for all timesteps to compute one by timestep. The CSieve can run on the modified method. The original and modified algorithms are shown in Figure 7.

Next, the closest approach determination computes all pairs that pass the filter in each time interval. The paralleling in this part depends on the number of pairs that pass the filter. The memory is allocated for relative distance on the closest approach of those pairs. Then, the collision probability is calculated for pairs with a relative distance that is less than the critical criteria. Lastly, the lists of conjunction data are recorded as the results.

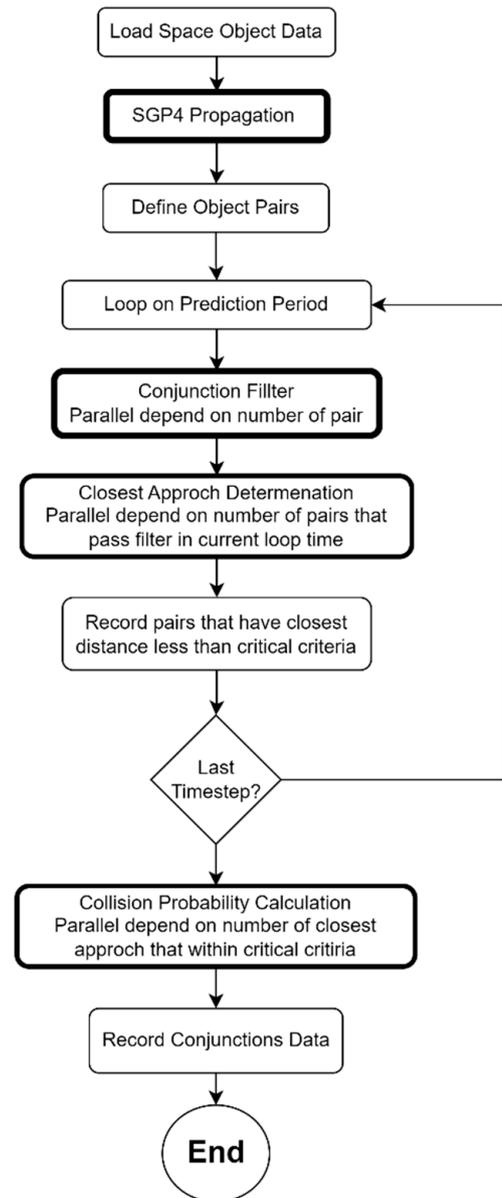
The GPU process is performed to analyze the conjunction of all space objects (24,023 objects) for one day. The prediction period begins on 3 March 2022 at 00:00:00 UTC with a timestep of one minute and sets the critical criteria of 5 km. The process runs on an Ubuntu server with Intel® Xeon® Gold 5218 CPU. The server is additionally equipped with an NVIDIA Tesla T4 GPU (2560 CUDA cores, 16 GB memory). The GPU process is tested and compared to the CPU single cores process on the same machine. All data in the GPU process are allocated in a type of double precision. Then, the conjunction data are not different from the CPU process. The runtime of GPU and CPU are recorded to observe the speedup of each filter when running on GPU. The percent of passed filters is calculated by the screening ratio of all vs. all processes. The investigated results of the overall screening of each filter method are summarized and discussed in Section 4.2.

Original



(a)

Modified



(b)

Figure 7. Implementation parallel computation with conjunction filter to analyze conjunction events. (a) parallel process standard for Smart Sieve and CAOS-D (b) modified parallel process for CSieve.

4. Results

4.1. One vs. All Results

This section shows the results of the one vs. all conjunction assessment. The five satellites are analyzed to define all possible conjunction events in advance seven days with three filter techniques. The results of all filters are compared with the non-filter process for accuracy performance as shown in Table 3.

In terms of accuracy, CSieve and CAOS-D provide the accuracy of analyzed results as same as the analyzed result without a filter process. The errors of Smart Sieve have been found because the timestep is unsuitable which leads to missing the time interval of the conjunction events. In order to confirm this assumption, we vary the timestep from one to ten minutes and investigate the conjunction events on each timestep of THEOS. Figure 8

shows that there are some errors in conjunction events of all variable timesteps of the Smart Sieve filter. Therefore, the assumption is correct.

Table 3. Number of conjunction events of the process with each filter on timestep one minute.

	THEOS	NAPA-1	NAPA-2	COSMOS 1675	IMAGE
Non-filter	21	13	9	4	1
Smart Sieve	18	11	8	4	1
CSieve	21	13	9	4	1
CAOS-D	21	13	9	4	1

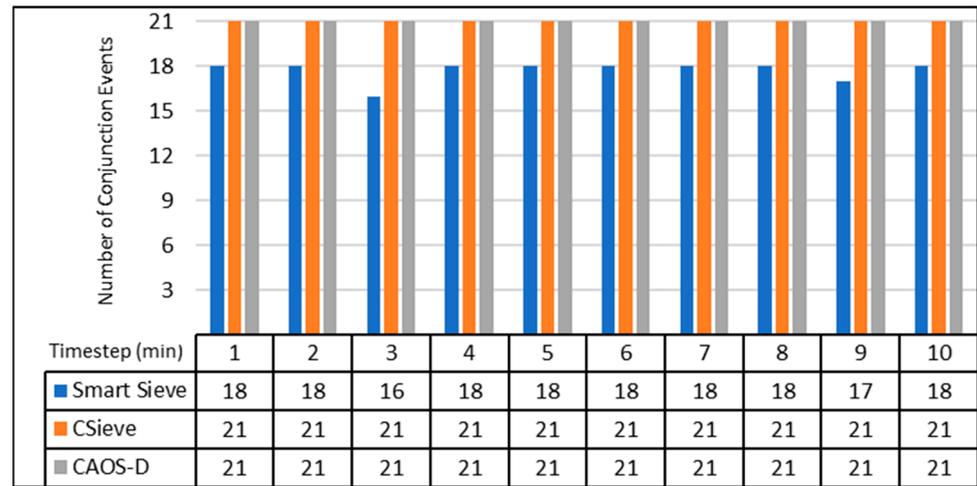


Figure 8. Number of Conjunction Events of THEOS on process with three filter by increasing timestep one to ten minutes.

In term of computation time, as shown in Figure 9, the computation times of Smart Sieve and CAOS-D continuously reduces when the timestep is increased because the increasing timestep reduces the total time interval leading to a reduced screening ratio (S_{pass}/S_{total}) of both Smart Sieve and CAOS-D as shown in Table 4. In contrast with the CSieve filter, the screening ratio is proportional to the timestep that is more than 2 min.

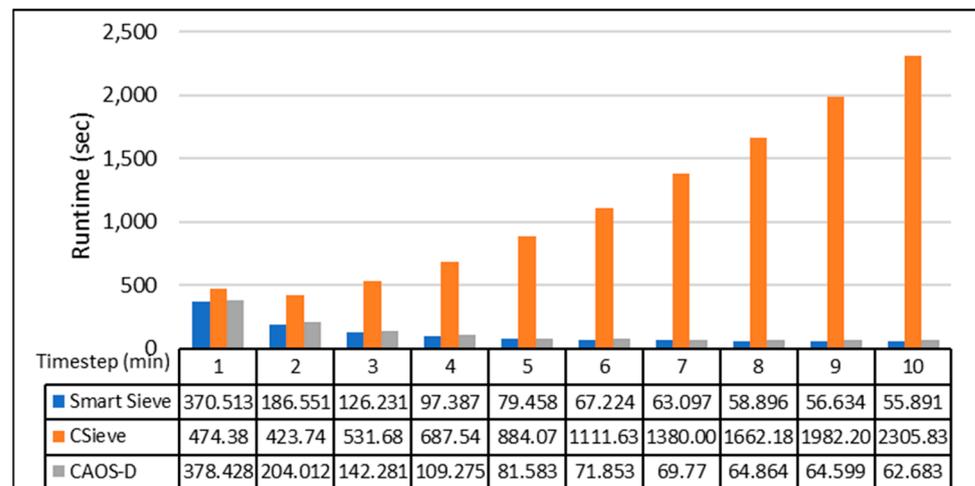


Figure 9. Process runtime of THEOS conjunction analysis with three filter methods by increasing timestep one to ten minutes.

Table 4. Several time intervals that pass filters of THEOS conjunction result by increasing the timestep one to ten minutes.

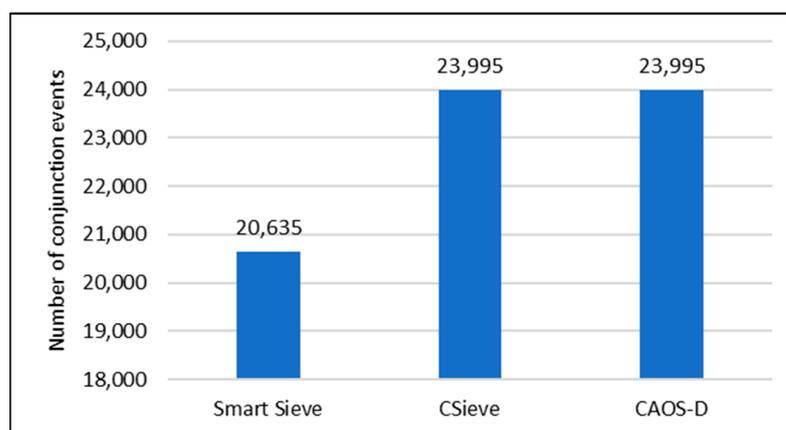
Timestep (Minute)	Total Screening (S_{total})	Number of Time Interval that Passes Filter (S_{pass})		
		Smart Sieve	CSieve	CAOS-D
1	222,949,440	1040	732,746	1188
2	111,474,720	4067	879,654	5106
3	74,316,480	5968	1,005,049	7524
4	55,737,360	7585	1,117,750	9740
5	44,589,888	9340	1,221,969	11,866
6	37,158,240	11,031	1,319,441	13,852
7	31,849,920	12,530	1,410,962	15,494
8	27,868,680	14,284	1,497,392	17,272
9	24,772,160	15,850	1,578,486	18,664
10	22,294,944	17,675	1,655,997	20,053

Smart Sieve and CAOS-D effectively eliminate some time intervals that pass the filter. However, Smart Sieve has a screening error by eliminating the time interval that yields the conjunction before determining the closest approach. If the Smart Sieve condition eliminates the time interval with minimum relative distance within critical criteria, it will miss a conjunction event. CAOS-D has a time interval that passes more than Smart Sieve, which can ultimately define the conjunction of five satellites in all timestep sizes. The Smart Sieve condition combined in CAOS-D (Minimum Distance and Fine Detection) can significantly screen the number of pairs that pass to the closest approach determination.

4.2. All vs. All Results

The three filter techniques are implemented on the GPU algorithm. Smart Sieve and CAOS-D methods run on the process standard while CSieve runs on the modified process with a timestep at one minute and a smaller number of time intervals that pass the filter.

In Figure 10, CSieve and CAOS-D can define the number of conjunction events as equal to 23,995 conjunction events but Smart Sieve can detect 20,635 conjunction events due to the same reason of the investigation in Section 4.1. The computation time of the combination conjunction filter and parallel techniques on the GPU in Figure 11 is significantly faster than running with a CPU single core. Smart Sieve and CAOS-D can reduce the computation time with a GPU by around two times. The computation times of CSieve can decrease from consuming 1,320.32 min or around 22 h to 58.29 min, which enhances the speed to 22.6 times; this method has a high screening ratio compared to other methods. It causes memory to be insufficient on GPU. However, the number of conjunction events of CAOS-D results are equal to CSieve results while taking less screening ratio. We summarize speedup and percent of screening ratio in Table 5.

**Figure 10.** Conjunction events by three filter method on all vs. all conjunction analysis with one-minute timestep.

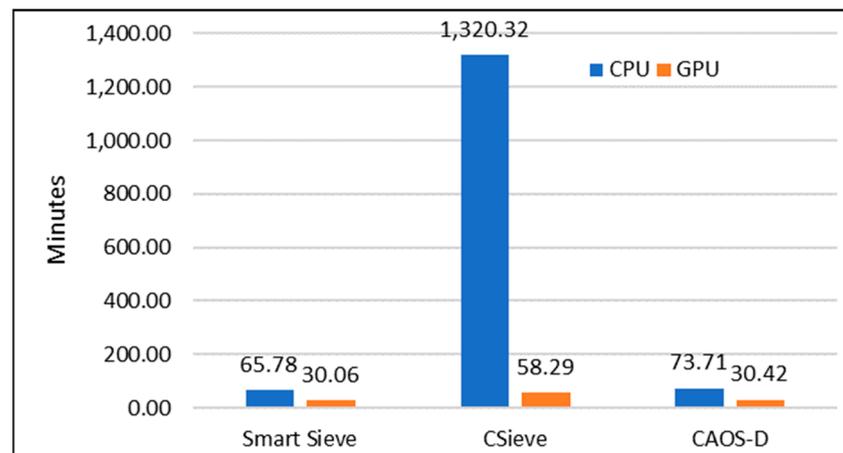


Figure 11. Computation time of filters on CPU and GPU.

Table 5. Speedup and percent of screening ratio.

Filter Method	Speedup	Percent of Screening Ratio
Smart Sieve	2.1×	0.00027%
CSieve	22.6×	0.26459%
CAOS-D	2.4×	0.00031%

5. Conclusions

This paper presents the performance investigation of the three conjunction filters: Smart Sieve, CSieve and CAOS-D techniques to enhance the computation speed of conjunction assessment analysis with GPU parallel computing. The investigation dataset of space objects is based on the space track website. The paper presents the analyzed results to identify the conjunction of a single satellite (one vs. all) and all space objects (all vs. all) with three filter techniques and validate the performance with the process without a conjunction filter to investigate the screening accuracy. GPU parallel computing can significantly decrease the computation time of all the filters. The screening accuracy of CSieve and CAOS-D can maintain the accuracy while Smart Sieve misses some conjunction events. CSieve can speed up computation more than other filters on GPU execution but CSieve is not suitable for parallel computation because a high screening ratio led to high computation and memory resource requirements. The combination method of CAOS-D has a low percentage of screening ratio and can maintain screening accuracy. The future work proposes to investigate the screening accuracy and computation time of three parameter combinations: screening filter, parallel computation and deep learning.

Author Contributions: Conceptualization, P.S., S.C. and S.S.; methodology, P.S., S.C. and T.L.; software, P.S. and K.P.; validation, S.C. and T.L.; formal analysis, P.S. and S.C.; investigation, K.P., P.S. and S.C.; data curation, P.S. and S.C.; writing—original draft preparation, P.S.; writing—review and editing, P.S., T.L. and S.C.; visualization, P.S.; supervision, T.L. and S.C.; project administration, S.C.; funding acquisition, S.C. and S.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Geo-Informatics and Space Technology Development Agency.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

This appendix shows the results of conjunction data that is compared with SOCRATES for validate algorithm.

Table A1. Compared conjunction data with SOCRATES.

Primary/Secondary	Conjunction Data	SOCARTES	Our Process	Difference
THEOS (33396)/CZ-4 DEB (26126)	TCA	24 March 2021 0:28:44.053	24 March 2021 0:28:44.056	0.003
	Relative distance (km)	0.675	0.69	0.015
THEOS (33396)/NOAA 16 DEB (41235)	TCA	24 March 2021 7:11:14.717	24 March 2021 7:11:14.721	0.004
	Relative distance	3.458	3.464	0.006
THEOS (33396)/FENGYUN 1C DEB (30918)	TCA	25 March 2021 17:22:01.420	25 March 2021 17:22:01.429	0.006
	Relative distance (km)	4.260	4.275	0.015
(44885) FLORIPASAT-1/PEGASUS DEB (39314)	TCA	24 March 2021 19:39:42.967	24 March 2021 19:39:42.972	0.005
	Relative distance (km)	0.023	0.0235	0.005
(25682) LANDSAT 7/FENGYUN 1C DEB (31741)	TCA	25 March 2021 23:48:18.545	25 March 2021 23:48:18.556	0.011
	Relative distance (km)	3.900	3.909	0.009

References

- Kessler, T.S.; Anz-Meador, D.P.; Matney, M.J. Space debris. *Int. Astron. Union Colloq.* **1996**, *150*, 201–208. [\[CrossRef\]](#)
- Kelso, T.S. Analysis of the Iridium 33-Cosmos 2251 collision. *Adv. Astronaut. Sci.* **2009**, *135*, 1099–1112.
- Kessler, D.J.; Cour-Palais, B.G. Collision frequency of artificial satellites: The creation of a debris belt. *J. Geophys. Res. Space Phys.* **1978**, *83*, 2637–2646. [\[CrossRef\]](#)
- ESA Space Debris Office. ESA'S Annual Space Environment Report. 2022. Available online: https://www.sdo.esoc.esa.int/environment_report/Space_Environment_Report_latest.pdf (accessed on 10 October 2022).
- Rodriguez, J.R.A.; Fadrique, F.M.M.; Klinkrad, H. Collision Risk Assessment with a Smart Sieve Method. In Proceedings of the Joint ESA-NASA Space-Flight Safety Conference, ESTEC, Noordwijk, The Netherlands, 11–14 June 2002; Volume 486, p. 159.
- George, E.R. A High Performance Conjunction Analysis Technique for Cluster and Multi-Core Computers. In Proceedings of the Advanced Maui Optical and Space Surveillance, Maui, HI, USA, 19–22 September 2011.
- Abernathy, B.; Surka, D.; Harvey, S.; O'Connor, M. *The CAOS-D Architecture for Conjunction Analysis*; Infotech@Aerospace: St. Louis, MO, USA, 2011; pp. 1434–1444.
- Fehr, M.; Navarro, V.; Martin, L.; Fletcher, E. The Process of Parallelizing the Conjunction Prediction Algorithm of ESA's SSA Conjunction Prediction Service Using GPGPU. In Proceedings of the 6th European Conference on Space Debris, Darmstadt, Germany, 22–25 April 2013; Volume 723, p. 152.
- Lin, M.; Xu, M.; Fu, X. A Parallel Algorithm for the Initial Screening of Space Debris Collisions Prediction using the SGP4/SDP4 models and GPU Acceleration. *Adv. Space Res.* **2017**, *59*, 2398–2406. [\[CrossRef\]](#)
- Stevenson, E.; Rodriguez-Fernandez, V.; Urrutxua, H.; Camacho, D. Benchmarking deep learning approaches for all-vs-all conjunction screening. *Adv. Space Res.* **2023**. [\[CrossRef\]](#)
- Sridharan, R.; Pensa, A.U.S. Space Surveillance Network capabilities. In Proceedings of the SPIE's International Symposium on Optical Science, Engineering, and Instrumentation, San Diego, CA, USA, 19–24 July 1998; Volume 3434.
- Healy, L.M. Close Conjunction Detection on Parallel Computer. *J. Guid. Control. Dyn.* **1995**, *18*, 4. [\[CrossRef\]](#)
- Alfano, S. Relating Position Uncertainty to Maximum Conjunction Probability. *J. Astronaut. Sci.* **2005**, *53*, 193–205. [\[CrossRef\]](#)
- Alfano, S. Determining a Probability-Based Distance Threshold for Conjunction Screening. *J. Spacecr. Rocket.* **2013**, *50*, 686–690. [\[CrossRef\]](#)
- Vallado, D.; Crawford, P.; Hujsak, R.; Kelso, T.S. Revisiting Spacetrack Report #3. In Proceedings of the Models for propagation of the NORAD Element Sets, AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Keystone, CO, USA, 21–24 August 2006; pp. 1–88.
- Rich, A. Investigating Analytical and Numerical Methods to Predict Satellite Orbits Using Two-Line Element Sets. Master's Thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, Dayton, OH, USA, 2017.
- Kelso, T.S.; Alfano, S. Satellite Orbital Conjunction Reports Assessing Threatening Encounters in Space (SOCRATES). In Proceedings of the 15th AAS/AIAA Space Flight Mechanics Conference, Vail, CO, USA, 9–13 August 2015.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.