# scientific reports

### OPEN



## Reducing the clustering challenge in the IoT using two disjoint convex hulls

Huxiong Li<sup>1</sup>, Bahram Sadeghi Bigham<sup>2</sup>, Mehdi Gheisari<sup>1,3,4,5</sup>, Aminreza Karamoozian<sup>1,6</sup>, Panjun Sun<sup>1</sup> & Yi Wan<sup>1</sup>

Accurate clustering of IoT devices is a promising challenge. We have observed that a few studies have been performed to address this challenge. However, they are expensive or do not shape accurate clustering. To fill this gap, in this study, we first solve a geometric version of a big challenge in pure mathematics: the NP-hard "Almost 2 - SAT" problem. Then, we solve it in a polynomial time. To clarify the concept, we present it as the "Two Disjoint Convex Hulls" challenge. We solve this challenge using two algorithms: the first is "Naive" and the second is faster than the "Naive" one can solve it in polynomial order,  $O(n^2)$ . In addition to providing a mathematical proof of our solution, we demonstrate its superior performance within an IoT industrial ecosystem.

**Keywords** Algorithm, Binary sensor network, Computational geometry, Convex hull, SAT, Separation axis theorem

Simple and low-cost sensors with limited processing power have gained significant popularity, especially in industrial and commercial systems, notably in Internet of Things (IoT) and Industrial IoT applications. These sensors are often used for detecting specific conditions and environmental monitoring, such as patient health monitoring<sup>1–3</sup>, early fire detection<sup>4</sup>, agriculture<sup>5,6</sup> or UAV recognition<sup>7</sup>. One of the main advantages of these sensors is their low price and scalability, allowing for widespread deployment to create a network of precise data collection points<sup>8,9</sup>.

Sensors (e.g., DHT11 and MQ-2) are well-suited for simple and low-cost applications in IoT systems, providing reliable environmental data for temperature, humidity, and gas detection. These sensors are extensively used in healthcare monitoring, industrial applications, and fire detection systems. For example, heart rate and blood oxygen sensors (e.g., MAX30100) are used in healthcare systems for real-time patient monitoring. The system employs ESP8266 microcontrollers and Arduino boards to connect the sensors to networks for data transmission and processing<sup>10</sup>. These components are ideal for large-scale IoT projects, offering scalability to manage numerous sensors while maintaining low power consumption and efficient data communication. Block diagrams and flowcharts are included to illustrate the sensor-to-gateway communication flow and data aggregation, ensuring a clear understanding of the system's architecture and its ability to scale effectively across various applications<sup>11–13</sup>.

Low-cost sensors and wireless networks play a critical role in healthcare systems, particularly for real-time localization and tracking of patients and medical equipment<sup>14</sup>. These technologies offer significant advantages in large hospitals and clinics, where efficient resource management and patient monitoring are crucial<sup>15</sup>. By deploying sensors or wireless tags, healthcare providers can track patients' movements within the facility, ensuring timely interventions when needed. For instance, patients with Alzheimer's conditions can be monitored and localized. Additionally, these systems are used for tracking essential equipment such as hospital beds, oxygen tanks, and wheelchairs. By leveraging low-cost sensors, healthcare facilities can improve efficiency, particularly in managing the movement of patients and medical assets in real-time.

Through observation, we have found that constructing corners of the shapes forging to reconstruction of binary images with disjoint components is an important problem<sup>16,17</sup>. This challenge can address many

<sup>1</sup>Institute of Artificial Intelligence, Shaoxing University, Zhejiang, China. <sup>2</sup>Department of Computer Science, Faculty of Mathematical Sciences, Alzahra University, Tehran, Iran. <sup>3</sup>Department of computer science and Engineering, Saveetha School of engineering, Saveetha Institute of Medical and Technical Science, Tamil Nadu, India. <sup>4</sup>Department of Computer Engineering, Shiraz Branch, Islamic Azad University, Shiraz, Iran. <sup>5</sup>Department of R&D, Shenzhen BKD Co LTD, Shenzhen, China. <sup>6</sup>Centre for Augmented Intelligence and Data Science, School of Computing, University of South Africa, Johannesburg, South Africa. <sup>III</sup>email: mehdi.gheisari61@gmail.com; Aminreza.karamoozian@yahoo.com challenges in different fields such as accurate clustering of IoT devices in Industrial IoT (IIoT) ecology, digital/ graphical games and so on. For all these applications, if we check them carefully, we would find that the main issue stems from the fact of detecting the minimum number of noises which could be removed and caused construction of some separate components.

To achieve our goal, as Fig. 1 illustrates, we consider an Industrial IoT environment where IIoT devices have spread in a field that tracks a UAV. In our scenario, IIoT devices are represented by + and - symbols, indicating the longitude and latitude of the moving autonomous vehicle (AV), respectively. In our scenario, two Convex Hulls have been created due to the existence of two kinds of IIoT devices. These convex hulls overlap, which also has a negative effect on the clustering issue of IIoT devices. They, generally, are without much complexity in a way that different sensing aspects can be applied in these systems to various targets such as temperature, light, sound, etc. The devices which generate information as little as one bit at each time result in providing inexpensive communication.

The goal is to minimize the number of IIoT devices required to achieve disjoint Convex Hulls, enabling accurate clustering.

In our scenario, each IIoT device detects and transfers one bit of information to the base station. These kinds of problems can be formulated as follows<sup>18</sup>. Having a set of *m* binary devices  $S = \{S_1, S_2, ..., S_m\}$  with the consideration of a single sample  $s \in \{-1, +1\}^m$  of data, produced at time *t*. The following lemma illustrates an approximation for the location of the tracked UAV which is outside the convex hulls of both plus devices and minus devices.

**Lemma 1** Let  $s \in \{-1, +1\}^m$  be a sample of the device values and X(t) be the location of the target UAV at the time t. Let  $A = \{S_i | s_i = +1\}$ ,  $B = \{S_j | s_j = -1\}$ , and CH(A) and CH(B) their convex hulls. Then  $CH(A) \cap CH(B) = \emptyset$ . Furthermore,  $X(t) \notin CH(A) \cup CH(B)^{18}$ .

This application enables the definition of new geometric problems, the logical versions of which are NP - hard. The "Almost 2SAT" problem, which serves as the foundational problem in this paper, is NP-hard. However, we do not attempt to solve this problem in its general form. The main contribution of this paper is the definition and solution of its geometric variant. In this work, we prove that when the problem is constrained to geometric data, it no longer remains computationally intractable, and we are able to solve it by a polynomial-time algorithm. Furthermore, we present two algorithms and discuss their details.

Section "Two disjoint convex hulls" includes some subsections which represents results<sup>19</sup> about these logical problems. In the next subsection, a new problem related to the convex hull is introduced, which is applicable to shape modeling and the IIoT environment, which we will solve it in polynomial time. As a result, a naive  $O(n^3)$  algorithm is also presented, and in the next part, to address this issue, we propose a novel solution which is  $O(n^2)$ . Finally, Section "Conclusion and future work" concludes this study and suggests several open problems.

#### Two disjoint convex hulls

As it is mentioned before, this problem is similar to a version of 2 - SAT problem, which would be discussed in the following.

#### Variable deletion almost 2-SAT

There are some basic hard problems in computational complexity, which are NP - hard. Although 2 - SAT problem belongs to class P, some versions of that like "Max 2 - SAT", "Almost 2 - SAT" and "Variable Deletion Almost 2 - SAT" are all NP - hard. The last problem is known in theoretical computer science under the terms Almost2 - SAT, All - but - k 2 - SAT, 2 - CNF deletion, and 2 - SAT deletion<sup>20</sup>.

In the "Almost 2 - SAT" problem, there are a 2 - CNF formula  $\phi$ , an integer k, and the question of whether one can delete at most k clauses from  $\phi$  to make it satisfiable. In "Variable Deletion Almost 2 - SAT" (variable-deletion variant), instead, it is allowed to detect at most k variables. Each variable is deleted along with all clauses containing it. One can think of deleting a variable as setting it both to true and false at the same time. These two problems are NP-hard, and in this regard, the following lemma has been proved<sup>19,21</sup>.



Fig. 1. IIoT ecology where devices have shaped two Disjoint Convex hulls with overlap.

**Lemma 2** Variable Deletion Almost 2 - SAT can be solved in  $4^k n^{O(1)}$  time.

#### The main problem

Detecting and testing the intersection between objects are among the most important applications of computational geometry in the Network domain. This problem has many other applications in text mining, and NLP in which we need to separate the given data set into some disjoint subsets<sup>22</sup>. It is one of the main questions addressed in Shamos' article that lays the groundwork for computational geometry<sup>23</sup>, the first application of the plane sweep technique<sup>24</sup>. It is hard to overstate the importance of finding efficient algorithms for intersection testing or collision detection as this class of problems has many applications in Robotics, Computer Graphics, Computer-Aided Design, and VLSI design<sup>25–27</sup>. In the plane, Shamos<sup>23</sup> presents an optimal linear algorithm to construct the intersection of a pair of convex polygons. Another linear time algorithm is later presented by O Rourke et al.<sup>28</sup>.

For the testing version of the problem, Lemma 3 has been proved<sup>29</sup>.

**Lemma 3** Let P and Q be two convex polygons with n and m vertices, respectively. The 2D-algorithm determines if P and Q intersect in  $O(\log n + \log m)$  time<sup>29</sup>.

As discussed earlier, the problems are applicable to binary sensor networks and target tracking. Assume that some of the sensors do not work properly in a binary sensor network; these sensors send incorrect signs. Clearly, if two convex hulls overlap, it means that some of the sensors are making mistakes (Fig. 1), but the reverse is not necessarily right.

In some applications, the minimum sensors are ignored because they are prone to make a wrong sign at the moment. This problem is summarized as Problem 1.

**Problem 1** *Minimum Sign to Remove* There are a plus set  $P_1$  and a minus set  $P_2$  of points on the plane. For an integer  $K \le min\{|P_1|, |P_2|\}$ , are there K points in  $P_1 \cup P_2$  whose removing yields two disjoint convex hulls for  $P_1$  and  $P_2$ ?

At first glance, it seems that the problem is NP - hard and structurally very similar to Variable Deletion Almost 2 - SAT problem. For some geometric reasons, however, the problem can be solved in polynomial time. The following observations can be considered easily about two disjoint convex hulls of given points in the 2D plane.

**Observation 1** For two given sets *A* and *B* of points in Problem 1, the minimum points which should be removed might be located outside of  $CH(A) \cap CH(B)$  (Fig. 2).

**Observation 2** Removing each pair of points on convex hull forces to remove all the points between them (either clockwise or counterclockwise), which are on the convex hull (Fig. 3).

**Observation 3** If we know the regions containing only plus (minus) signs, the problem may become simpler (Fig. 4).

To decide whether two convex polygons are intersecting, the Separating Axis Theorem can be used<sup>30,31</sup>. Clearly, for two not-intersecting convex polygons, there exists a line passing between them. Obviously, such a line exists if and only if an edge of one of the polygons forms that line. The closest point of this line to the other polygon is one of its corners which is closest to the first polygon (Fig. 5). This edge will then form a separating axis between the polygons. And also, if two middle edges of two polygons are parallel, both of them are separating axes. These are summarized in Lemma 4, and the idea has been used in the algorithm of this study<sup>30,32</sup>.

**Lemma 4** (Hyperplane Separation Theorem) Let A and B be two disjoint nonempty convex subsets of  $\mathbb{R}^n$ . Then there exist a nonzero vector v and a real number c such that  $\langle x, v \rangle \geq c$  and  $\langle y, v \rangle \leq c$  for all x in A and all y in B; i.e., the hyperplane  $\langle ., v \rangle = c$ , v the normal vector, separates A and  $\mathbb{B}^{30}$ .



Fig. 2. Removing only two (minimum) IIoT devices yields two disjoint convex hulls.

Scientific Reports | (2025) 15:17200



Fig. 3. Removing points (a) and (d) forces to remove either (e) or (b, c).





For every two arrays  $A_{2*n}$  and  $B_{2*n}$  of points in the plane, Algorithm 1 computes a cost for each line segment made by a pair of points. It chooses the pair with the minimum cost and returns the points that should make the separating axis. As a result, the minimum number of removing signs (plus or minus) of Problem 1 will be represented. So, a glance at the algorithm reveals that Theorem 1 is clear.





**Input:** *n* points with plus sign  $P = \{p_1, p_2, ..., p_n\}$  and *m* points with minus sign  $M = \{ m_1, m_2, ..., m_m \}$  in the plane. **Output:** cost *c* and Indexes *i* and *j* which minimizes cost[P, M].1:  $c = \infty$ ,  $a = p_1$ ,  $b = m_1$ ; 2: **for** *i* = 1, ..., *n* **do** 3: **for** *j* = 1, ..., *m* **do** if  $p_i$  and  $m_j$  are the same points then 4:  $cost[i, j] = \infty$ 5: else 6: rightp[i, j] = Number of plus points in P that 7: are at the right hand side of line *ij*; leftp[i, j] = Number of plus points in *P* that are 8: at the left hand side of line *ij*; 9: *rightm*[*i*, *j*]= Number of minus points in *M* that are at the right hand side of line *ij*; *leftm*[*i*, *j*] = Number of minus points in *M* that 10: are at the left hand side of line *ij*; cost[i, j]=min{(rightp[i, j] 11: +leftm[i, j], (rightm[i, j] + leftp[i, j]); end if 12: 13: if  $cost[i, j] \leq c$  then c = cost[i, j], a = i, b = j;14: end if 15: end for 16: 17: end for 18: Return *c*, *a*, *b*;

Algorithm 1. A Naive Algorithm for Two Disjoint Convex Hulls

**Theorem 1** Algorithm 1 solves Two Disjoint Convex Hulls in  $O(m+n)^3$ .



Fig. 6. Primal Plane, red points  $\{r_1, r_2, ..., r_7\}$  and blue points  $\{b_1, b_2, ..., b_8\}$ .



**Fig.** 7. Vertex  $V_1$  is assigned to the starting cell in the upper envelope.

#### An $O(m+n)^2$ algorithm

As mentioned earlier, in this section, a new fast algorithm is presented for the problem that uses the duality of given points. As illustrated in Fig. 6, if the dual space is drawn,  $(m + n)^2$  regions appear in which m = |A| and n = |B|. In order to find the optimal line  $l_{opt}$  in the primal space, an optimal cell *cell<sub>opt</sub>* and a point  $v_{opt}$  in it should be found in dual space that  $v_{opt}$  demonstrates the dual of  $l_{opt}$  which has been depicted in Fig. 9.

For this purpose, firstly, the dual line of all the given points in A (plus signs) and B (minus signs) is drawn in the dual space. Red and blue colors are assigned to dual lines of set A and B, respectively (Lines 2–3). There appear  $O(m + n)^2$  segments and half-lines in two colors in the dual space. The dual space is stored in a Doubly Connected Edge List (DCEL) in which the edge list contains the color of every edges as well. This process takes  $O(m + n)^2$  time. At this time, a planar graph G(V, E) (Lines 4–6) can be generated as follow. The starting cell to which vertex  $v_1$  is assigned to that is the cell above the upper envelope (Fig. 7). A vertex  $v_i$  is assigned to every other cell in any order. There is an edge between two vertices if those are adjacent, so E contains all the edges between every two adjacent vertices (Line 6) (Fig. 8). The resulting graph has  $O(m + n)^2$  vertices and edges, each of which processes once (Fig. 9).

The next step is computing a pair of numbers  $w_1$  and  $w_2$  as weights of the vertices (7–25). Firstly, set  $w_1(v_1) = m$  and  $w_2(v_1) = 0$ , then set weights of the adjacent vertices according to the segment between them, so that if the segment is red decrease  $w_1$  without any change in  $w_2$  and if the segment is blue reduce  $w_1$  and w - 1 stay without change. Therefore, for each vertex v,  $w_1(v)$  and  $w_2(v)$  show the number of red lines under v and the number of blue lines above it respectively. Using a queue structure Q, the weight for all the vertices (Lines 8–25) is computed. In each step, when an unweighted vertex appears by crossing a red (blue) segment,  $w_1(w_2)$  is decreased (increased) by one. Computing the weight for a new vertex takes constant time; hence, in  $O((m + n)^2)$  time, the weighted planar graph can be computed in which the weights  $w_1(v)$  and  $w_2(v)$  demonstrate the red lines below and blue lines above the cell containing vertex v. In both sensor network applications and finding two



Fig. 8. Every other cell is labeled in any order and there is an edge between two vertices if and only if their corresponding cells are adjacent.



**Fig. 9**. Dual plane of Fig. 6, dual of points (lines) and a black point in a region that separates blue and red lines with minimum error (only  $D(b_7)$  and  $D(r_8)$  are at the wrong side of black point.

disjoint convex hulls, if the separation line is the dual line of vertex v, then  $w_1(v)$  plus signs and  $w_2(v)$  minus signs should be removed. Algorithm 2 find the optimal vertex  $v_{opt}$  with  $min\{w_1 + w_2\}$  and line  $l_{opt}$  in Lines (26–27) in  $O((m + n)^2)$  time. We can conclude all the results in Theorem 2.

**Theorem 2** Let A and B be two sets of points in the plane, |A| = m and |B| = n. Then Algorithm 2 finds the minimum number of removal points in order to have two disjoint convex hulls in  $O(m + n)^2$ .

#### **Input:** Two set of *n* plus and *m* minus points

**Output:** Optimal separation line to have minimum number of removal points and two disjoint convex hulls.

- 1: Compute the dual line for every points in *A* and *B*;
- 2: Generate DCEL for the arrangement;
- 3: For each half-edges in DCEL assign label *r* (red for Plus points) or *b* (blue for minus points);
- 4: Find the upper envelope of arrangement;
- 5: Generate a planar graph G(V, E);
- 6:  $V = \{v_1, v_2, ..., v_{(m+n)^2}\}$  in which  $v_1$  stands for the cell above the upper envelope and  $v_2, ..., v_{(m+n)^2}$  for other cells;
- 7: Make *E* by connecting every two vertices which their cells are adjacent;
- 8: Set weight w<sub>1</sub>(v<sub>1</sub>) = m) and w<sub>2</sub>(v<sub>1</sub>) = 0, which m is the number of red lines.
- 9: Add  $v_1$  to empty queue  $Q_i$
- 10:  $V' = \{v_1\};$
- 11: **for** all item *u* in queue *Q* **do**
- 12: **for** all vertex *v* in *V* which their cell are adjacent to *u* **do**
- 13: **if** v is not in V' and the segment between *u* and *v* is red **then**

14: 
$$w_1(v) = w_1(u) - 1;$$

15: 
$$w_2(v) = w_2(u);$$

- 16: **else**
- 17: **if** v is not in V' and the segment between *u* and *v* is blue **then**
- 18:  $w_1(v) = w_1(u);$

19: 
$$w_2(v) = w_2(u) + 1;$$

- 20: end if
- 21: end if
- 22: Add v to V';
- 23: Add v to queue Q;
- 24: end for
- 25: Remove u from queue Q;
- 26: end for
- 27: Find the vertex  $v_{opt}$  in *G* with minimum  $w_1 + w_2$ ;
- 28: Return *l<sub>opt</sub>=dual(v<sub>opt</sub>)*, the optimal separation line in the prime space;

Algorithm 2. Two Disjoint CHs by Dual Space

#### Note to practitioners

This paper has been inspired by a real challenge in the world: imperfect IoT device detection. Specifically, the challenge stems from the fact that we need to leverage existing generated data of a variety of IoT devices to find imperfect ones. This problem literally is NP-hard which means there is no solution that can solve this issue in a logical polynomial time. In this paper, we have extracted the geometric version of IoT devices data forging an accurate and fast algorithm for it. We demonstrate that the geometric version of the problem is not NP-hard. To the best of our knowledge, this is the first study to address this complex challenge. Moreover, we address this geometric version through two novel fast algorithms for a large number of IoT devices in  $O(n^2)$ . The achieved result demonstrates we can address this important challenge in a short time. To show the superior performance of our solution, we have implemented it in an IoT environment and evaluated it from several evaluation metrics.

#### Conclusion and future work

In this paper, we addressed the accurate clustering issue in the Industrial IoT environment by addressing a big challenge which is "Almost 2 - SAT" problem. In this case, at first, we have depicted it in a geometric environment. Then, we solved it via two different algorithms, while the first one is a naive and simple but slower approach with  $O(n^3)$  time complexity. However, the second algorithm uses dual space and is faster with  $O(n^2)$  time complexity. Beyond mathematical proof, we showed the effectiveness of our solution in the Industrial IoT environment. We showed that through our solution, we could remove the minimum number of IoT devices in the Industrial IoT ecology and perform accurate clustering of IoT devices.

As highlighted throughout the paper, the two algorithms presented in this paper fall within class P and exhibit polynomial computational complexity, ensuring high execution speed and making them suitable for realtime applications. However, the system's scalability and adaptability to diverse healthcare scenarios could face limitations, particularly in environments with higher complexity or a larger number of IoT devices. While the algorithms perform well in real-time localization using low-cost devices, further optimization may be necessary to enhance their robustness and scalability in more demanding settings, such as large-scale hospital networks or disaster response scenarios. Future work will focus on improving system adaptability to various healthcare applications, optimizing the algorithms for even greater scalability, and addressing any constraints related to real-time performance. Additionally, future research may explore integrating advanced machine learning models to detect sensor malfunctions more effectively and expanding the system's applicability to other fields, such as drones, wildfire detection, and smart city infrastructure.

#### Data availability

All data generated or analysed during this study are included in this published article (data is validated through mathematical proof in this published paper).

Received: 17 May 2024; Accepted: 31 January 2025 Published online: 17 May 2025

#### References

- Kaur, B., Kumar, S. & Kaushik, B. K. Novel wearable optical sensors for vital health monitoring systems-A review. *Biosensors* 13(2), 181 (2023).
- 2. Yu, A. et al. Implantable flexible sensors for health monitoring. Adv. Healthcare Mater. 13(2), 2302460 (2024).
- Zhang, M. et al. A review of radio frequency identification sensing systems for structural health monitoring. *Materials* 15(21), 7851 (2022).
- 4. Sharma, A. et al. An IoT-based forest fire detection system: Design and testing. *Multimedia Tools Appl.* 83(13), 38685–38710 (2024).
- Luo, J., Zhao, C., Chen, Q. & Li, G. Using deep belief network to construct the agricultural information system based on Internet of Things. J. Supercomput. 78(1), 379–405. https://doi.org/10.1007/s11227-021-03898-y (2022).
- Gheisari, M. et al. An efficient cluster head selection for wireless sensor network-based smart agriculture systems. Comput. Electron. Agric. 198, 107105 (2022).
- 7. Karam, S. N. et al. Intelligent IoT-and UAV-assisted architecture for pipeline monitoring in OGI. IT Professional 26(3), 46-54 (2024).
- Li, T., Kouyoumdjieva, S. T., Karlsson, G., & Hui, P. Data collection and node counting by opportunistic communication. In *Paper presented at the 2019 IFIP Networking Conference (IFIP Networking) from (2019)*. https://doi.org/10.23919/IFIPNetworking46909. 2019.8999476
- Lu, J. & Osorio, C. On the analytical probabilistic modeling of flow transmission across nodes in transportation networks. *Transp. Res. Rec.* 2676(12), 209–225. https://doi.org/10.1177/03611981221094829 (2022).
- Sun, G. et al. Cost-efficient service function chain orchestration for low-latency applications in NFV networks. *IEEE Syst. J.* 13(4), 3877–3888. https://doi.org/10.1109/JSYST.2018.2879883 (2019).
- Elahi, H., Wang, G., Jiang, W., Bartel, A. & Le Traon, Y. A qualitative study of app acquisition and management. *IEEE Trans. Comput. Soc. Syst.* 11(2), 1907–1925. https://doi.org/10.1109/TCSS.2023.3288562 (2024).
- Sun, G., Li, Y., Liao, D. & Chang, V. Service function chain orchestration across multiple domains: A full mesh aggregation approach. *IEEE Trans. Netw. Serv. Manage.* 15(3), 1175–1191. https://doi.org/10.1109/TNSM.2018.2861717 (2018).
- Li, H. et al. LMFLS: A new fast local multi-factor node scoring and label selection-based algorithm for community detection. Chaos Solit. Fract. 185, 115126 (2024).
- 14. Pradhan, D. et al. Dynamic data placement strategy with network security issues in distributed cloud environment for medical issues: An overview. Recent Advances in Computer Science and Communications (2024).
- 15. Izadpanah, K. M. et al. AI in nuclear medical applications: Challenges and opportunities. *Front. Biomed. Technol.* **11**(2), 158–161 (2024).
- Wu, S., Zhang, S., Xu, D. & Huang, D. A weighting localization algorithm with LOS and one-bound NLOS identification in multipath environments. J. Inf. Sci. Eng. 35(6), 1209–1222. https://doi.org/10.6688/JISE.201911\_35(6).0003 (2019).
- Zhou, G., Zhou, X., Chen, J., Jia, G. & Zhu, Q. LiDAR echo gaussian decomposition algorithm for FPGA implementation. Sensors 22(12), 4628. https://doi.org/10.3390/s22124628 (2022).

- Aslam, J., et al. Tracking a moving object with a binary sensor network. In Proceedings of the 1st international conference on Embedded networked sensor systems. (ACM, 2003).
- 19. Cygan, M. et al. Parameterized algorithms 66-68 (Springer, Berlin, 2015).
- 20. Razgon, I. & O'Sullivan, B. Almost 2-SAT is fixed-parameter tractable. J. Comput. Syst. Sci. 75(8), 435-450 (2009).
- Kosari, S. et al. A hybrid discrete grey wolf optimization algorithm imbalance-ness aware for solving two-dimensional bin-packing problems. J. Grid Comput. 22, 49 (2024).
- 22. Aoyama, K., Saito, K. & Ikeda, T. CPI-model-based analysis of sparse k-means clustering algorithms. Int. J. Data Sci. Anal. 12(3), 229-248 (2021).
- 23. Shamos, M. I. Geometric complexity. In Proceedings Of The 7th annual ACM symposium on theory of computing, pp 224–233. (ACM, 1975)
- 24. Shamos, M. I. & Hoey, D. Geometric intersection problems. In *Proceedings of the 17th annual symposium on foundations of computer science*, pp 208–215. (IEEE, 1976).
- Toth, C. D., O'Rourke, J. & Goodman, J. E. Handbook of discrete and computational geometry (Chapman and Hall/CRC, Boca Raton, 2017).
- 26. Jiménez, P., Thomas, F. & Torras, C. 3D collision detection: A survey. Comput. Graph. 25(2), 269-285 (2001).
- 27. Lin, M. & Gottschalk, S. Collision detection between geometric models: A survey. In Proc. of IMA conference on mathematics of surfaces. Vol. 1. (1998).
- 28. O'Rourke, J. et al. A new linear algorithm for intersecting convex polygons. Comput. Graphics Image Process. 19(4), 384–391 (1982).
- 29. Barba, L., & Langerman, S. Optimal detection of intersections between convex polyhedra. In *Proceedings of the twenty-sixth annual* ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics (2015).
- 30. Boyd, S. & Vandenberghe, L. Convex optimization (Cambridge University Press, Cambridge, 2004).
- Gilbert, E. G., Johnson, D. W. & Sathiya Keerthi, S. A fast procedure for computing the distance between complex objects in threedimensional space. *IEEE J. Robot. Automat.* 4(2), 193–203 (1988).
- 32. Eberly, D. Intersection of convex objects: The method of separating axes. Geometric Tools, LLC https://www.geometrictools.com, (1998-2008) (2001).

#### Author contributions

Huxiong Li: supervision Bahram Sadeghi Bigham: Writing draft Mehdi Gheisari: Project management, Aminreza Karamoozian: Review and validation, Panjun Sun: conceptualization and methodology, Yi Wan: Funding allocation All authors reviewed the manuscript.

#### Declarations

#### **Competing interest**

The authors declare no competing interests.

#### Additional information

Correspondence and requests for materials should be addressed to M.G. or A.K.

Reprints and permissions information is available at www.nature.com/reprints.

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

© The Author(s) 2025