Mohammed Ali-Khan

CSE 5311.005

2 December 2024

Graham's Scan: A solution to the Convex Hull Problem

Abstract.

The Graham Scan is a computational geometry algorithm that specializes in computing the convex hull of a set of defined points. Its simplicity and versatility allows it to be used with multidimensional datasets, In a number of applications such as Robotics, Computer Graphics, and Data Analysis.

Introduction.

Modern computer graphics and visualization applications are made up of the basic need to connect two- and three-dimensional points together, creating a virtual representation of the physical world. Since the idea of such visual aids is so important for anything that needs to be seen, a computational algorithm is needed. Hence, the concept of Computational Geometry is used, as "Computational geometry is a field of study that focuses on developing algorithms and data structures for solving problems that involve geometric shapes and structures".

With computational geometry being an important topic in triangulation and diagrams, another important problem tackled by computational geometry is Convex Hulls. "A convex hull is the smallest convex polygon that contains a given set of points". One way to understand this problem is to consider the following example, "Imagine that the points are nails sticking out of the plane, take an elastic rubber band, stretch it around the nails and let it go. It will snap around the nails and assume a shape that minimizes its length. The area enclosed by the rubber band is called the convex hull …". There are a few solutions to computing the convex hull, such as Jarvis's March, Chan's algorithm and the Kirkpatrick-Seidel algorithm. However, a simpler algorithm for computing a convex hull is the Graham Scan.

The Graham Scan "is a simple and efficient algorithm for computing the convex hull of a set of points". It does so by iteratively adding points to the convex hull, checking the direction of the point in relation to the last point, and determining if the edge can be shortened. Once all the points have been checked, the result is the completed convex hull.

Background and History

The Graham Scan provides a solution to finding the convex hull. The convex hall is the smallest polygon for which each point is either on the boundary of the polygon or enclosed within the polygon. There are three assumptions made for the problem. One: All points are

unique. Two: The polygon consists of at least three points. And Three: At least three points are not colinear.

The Graham Scan has certain advantages when compared to other convex hull algorithms, such as the Jarvis March and the Divide and Conquer Approach. For example, when compared to the Jarvis March, the Graham Scan is faster with a time complexity of O(n log n) as opposed to Jarvis March's time complexity of O(n h) where h is the number of vertices. This is due to the sorting step that Graham Scan has, where sorting the points with respect to the lowest point, allows for the identification of points on the final convex hull to be much faster. When compared to the divide and conquer approach, the Graham Scan is better suited and built for two dimensional problems.

The Graham Scan is named after Ronald Graham, who published the original algorithm in 1972. In the publication, Graham outlines the steps necessary to use the algorithm to solve a convex hull. Graham's original description of the algorithm had the starting point be in the interior of the polygon, rather than the edges, which resulted in a polygonization, or some form of star shaped polygon of all the vertices. This algorithm later evolved into calculating a polygon around all the vertices, rather than with all the vertices.

Theory and Key Characteristics.

"The convex hull of a set of points is defined as the smallest convex polygon, that encloses all the points in the set. Convex means that the polygon has no corner that is bent inwards." To represent or solve the convex hull in code, the points are usually represented as a list of points.

The Graham Scan works with two main fundamental properties. Polar angle sorting determines the starting point of the scan. Typically, the point with the lowest y-coordinate is taken. If there are multiple points that lie on the same lowest y-coordinate, the point with the greatest x-coordinate value may be preferred. Next the angle between the starting point and all of the other points are calculated, and the points are sorted based on the angles. If a point is at the same angle in respect to the starting point, then the point with a shorter distance between it and the starting point is taken first. Hence, all of the points are sorted by their polar angle to the starting point.

The algorithm will then take every point in its sorted list and determine whether the point creates a concave or convex corner. The convex corner is what is needed for a convex hull. The way to categorize the corner as either a left turn or a right turn by using the cross product of the line vectors. The cross product is a vector multiplication process where the result of the multiplication consists of a direction and a magnitude. The direction of the cross product is determined by the right-hand rule, and in the case of the Graham Scan, determines in which direction the point in question lies. A point to the right of the scan would result in a positive direction of the cross product, while a point in the left would result in a negative direction of the cross product. Thus, a positive magnitude would indicate a concave corner, while a negative magnitude would indicate a convex corner. The final polygon of the convex hull would continue to be built with convex corners, but a concave corner indicates that the three points in question can be compressed into only two points to take out the concave point. The process repeats until only convex corners remain in the final polygon.

The correctness of this algorithm can be argued by the following methodology. Suppose after a convex hull is found, there exists a point outside of the convex hull. This indicates that the point had been removed since it along with its neighboring points created a concave corner. This point's neighboring points would still be in the border, implying that the point in question lies within this border, which forms a contradiction, as the point cannot be both inside and outside of the border at the same time. Thus, all points are either on the border, as they form convex corners with other neighboring points, or within the border as the hull encompasses them. Essentially, if a concave corner exists on the border of the hull, or a right turn is made going counterclockwise, the polygon can be reduced by removing the middle point and combining the other neighboring points making a convex corner, or a point with a left turn.

Implementation.

```python
from math import atan2


def ccw(p1, p2, p3):
    """
    Check if the movement from p1 to p2 to p3 makes a counter-clockwise turn.
    Returns:
    > 0: Counter-clockwise turn
    = 0: Collinear
    < 0: Clockwise turn
    """
    return (p2[0] - p1[0]) * (p3[1] - p1[1]) - (p2[1] - p1[1]) * (p3[0] - p1[0])


def graham_scan(points):
    # Find the point with the lowest y-coordinate and leftmost if ties (P0)
    P0 = min(points, key=lambda p: (p[1], p[0]))

    # Sort points by polar angle with P0, breaking ties by distance
    points.sort(key=lambda p: (atan2(p[1] - P0[1], p[0] - P0[0]), (p[0] - P0[0]) ** 2 + (p[1] - P0[1]) ** 2))

    # Use a stack to hold the points on the convex hull
    stack = []

    for point in points:
        # Remove points from the stack if they cause a clockwise turn
        while len(stack) > 1 and ccw(stack[-2], stack[-1], point) <= 0:
            stack.pop()
        stack.append(point)

    return stack


points = [(0, 4), (2, 9), (4, 3), (6, 0), (8, 6)]
convex_hull = graham_scan(points)
print("Convex Hull:", convex_hull)
```

Figure 1. Code implementation of Graham Scan

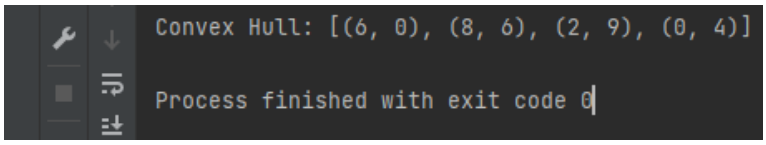
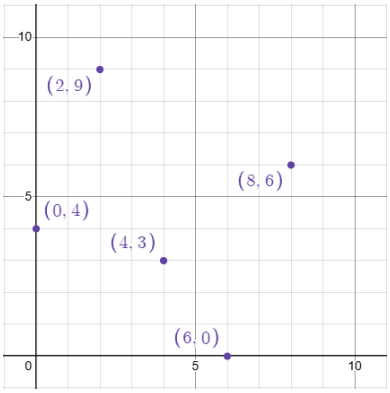
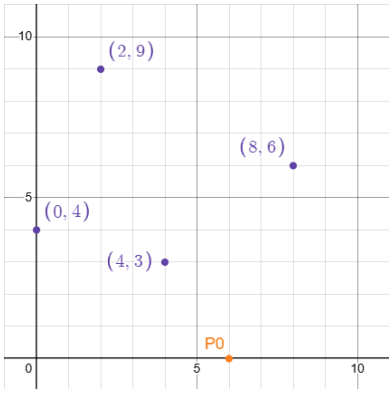


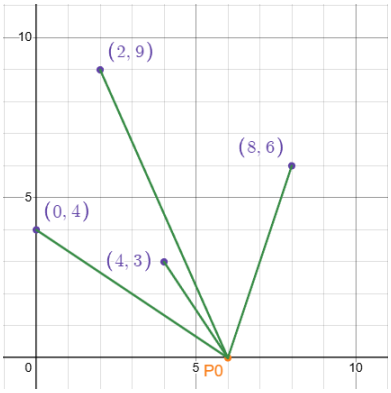
Figure 2. Result of Graham Scan python code. Visually explained in Figures 3-14

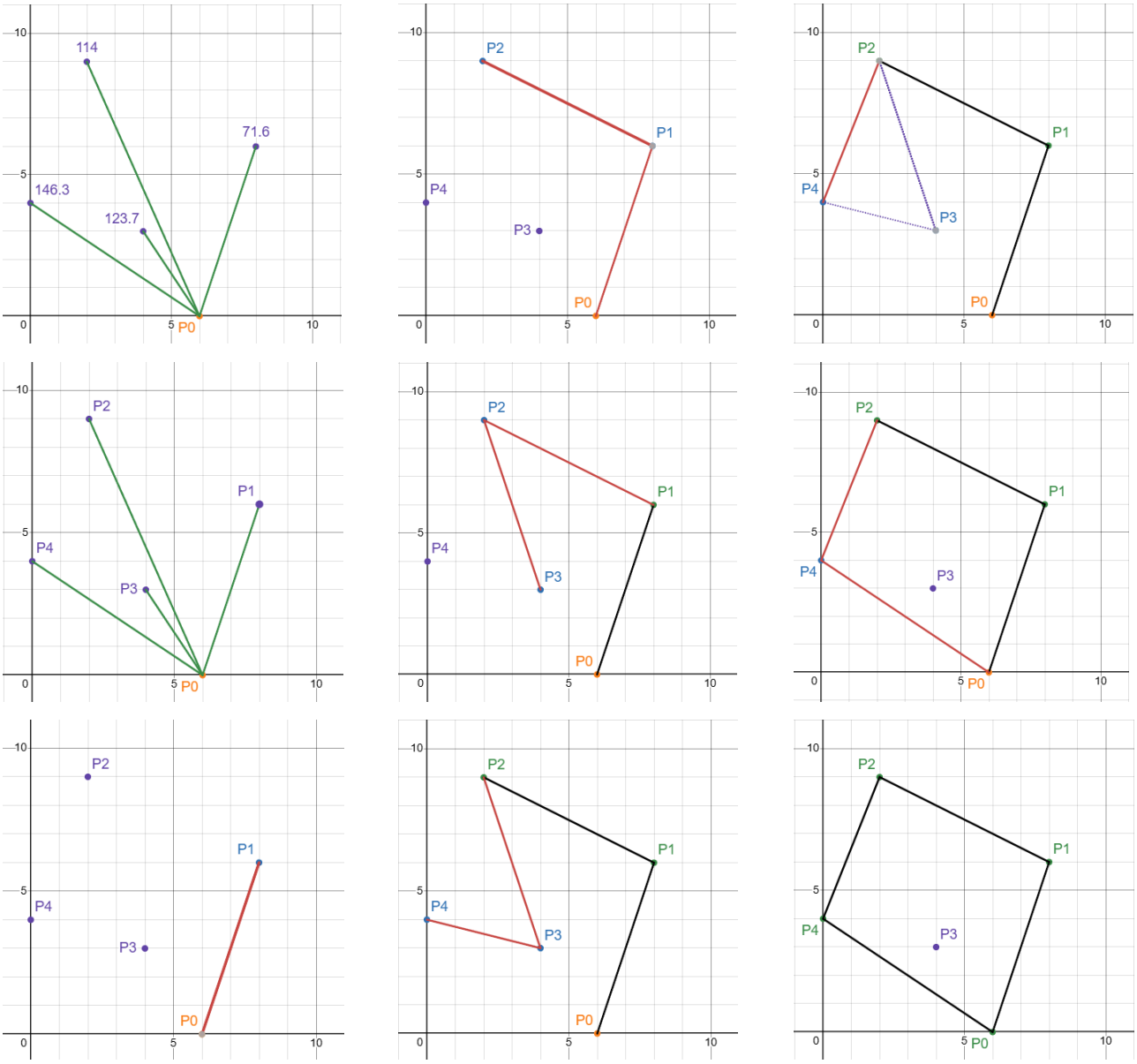
A set of points are considered, where any number of dimensions can be included, but for the simplest approach a two-dimensional set of points has x and y coordinates. With the points set, a starting point is needed, where typically the lowest point is chosen (the point with the smallest y value). The angles of the points in relation to the starting point need to be calculated, so that the points can be ordered. For visual understanding, a line segment is drawn between the points and starting point for the polar angle to be calculated. An arctan formula is used to solve for the polar angle. The points are then sorted, with points that have a shorter distance to the starting point having preference when the polar angle is the same.

A stack is then initialized, with the starting point and the next sorted point being pushed into the stack. A third point is added to the stack, this point being the next sorted point. The orientation on whether this corner creates a right turn, or a left turn is calculated. If the corner is a right turn, this indicates a concave corner, and the vertex is removed from the stack, and the scan rechecks to ensure the new hull continuously makes no right turns. If the corner is a left turn, or if removing the right turn vertex resulted in a left turn, the next point is added and the process repeats. The process ends when the last point is able to connect back to the starting point, while also creating a left turn, as a right turn would mean that last point has to be removed. In this way, the convex hull is the path of all of the points in the stack, as this stack now represents the points that lie on the smallest possible polygon that encloses all of the points.









Figures 3 – 14. The above figures show a visual representation of the Graham Scan in columnar order. 3) A set of points is given. 4) The lowest is set as the pivot. 5-6) The polar angle is calculated. 7) Points are sorted by polar angle. 8-9) first corner mapped. 10) second corner. 11) Third corner is a right turn. 12) P3 removed and new corner is added as second corner. 13) Third corner added, completing the loop. 14) Final Convex Hull.

Applications.

The Graham Scan is useful for real world applications, especially in the field of robotics and computer graphics. For example, Geographical Information Systems can use the Graham Scan for boundary determination, by integrating the data received from satellite imagery and field observations into precise boundaries used for geophysical surveys. Another application is obstacle mapping and navigation for robotics. For instance, a self-driving vehicle can use the

Graham scan to survey the obstacles in front of or around the vehicle, indicating that so as long as the vehicle remains within the boundary, it will not collide with such obstacles.

The Graham Scan is extensively used in Object rendering and boundary creation for computer graphic software. Mapping a three-dimensional object virtually onto a physical two-dimensional screen, such as in the case of computer animation or visual effects, would mean creating the boundary for those vertices to be seen at a certain artistic perspective. In this way, the Graham Scan would allow graphics engines to visualize the data to display into a rendered animation or image. Physics simulations and machine learning models can also use the Graham Scan by defining clear boundaries between clusters of data. This allows for analysts to have clearly defined boundary regions dependent on the attributes that result in them.

Benchmarking and Analysis.

The runtime of the Graham Scan is O(n log n) since sorting the points by their polar angle takes O(n log n), and going through each of the points to find right or left turns takes O(n). Thus, the complete runtime is O(n log n) + O(n) = O(n log n).

Challenges and Limitations.

Additionally, there is an anomaly when solving for the convex hull with collinear points. Suppose three points lie on the same line. The convex hull would consist of the two ends of the line as the point in the middle is encased by a one-dimensional polygon. Other implementations have all three points as part of the convex hull, since the polygon itself does in fact touch the middle point. All in all, whether the middle point is in or out of the convex hull solution depends on the problem. Others have also suggested that these problems can be ignored completely.

Conclusion.

In conclusion, the Graham Scan is an important algorithm within the field of computational geometry, that solves the problem of a convex hull. The algorithm is especially useful in the field of computer graphics and data analytics. The algorithm is efficient in its implementation, by sorting the points by polar angle and iteratively finding the convex path to enclosing all of the points in the smallest polygon possible. The Graham Scan is versatile in that it can be used for a multi-dimensional set, as well as be modified to include colinear points on the path. Its simplicity allows it be used in a variety of applications, and continues to be on the precipice of computational geometry.

Works Cited.

1. Abiy, Thaddeus, et al. “Convex Hull: Brilliant Math & Science Wiki.” *Brilliant*, brilliant.org/wiki/convex-hull/. Accessed 30 Nov. 2024.
2. Baker, Daniel W., and William Haynes. “Engineering Statics: Open and Interactive.” *Statics*, engineeringstatics.org/cross-product-math.html. Accessed 30 Nov. 2024.

3. Bpreston. “How Are the Boundaries of a Geophysical Survey Determined?” *MAJR Resources*, 13 Apr. 2024, majrresources.com/how-are-the-boundaries-of-a-geophysical-survey-determined/#:~:text=One%20of%20the%20main%20functions,of%20the%20area%20under%20survey.
4. Cajic, Dino. “Graham’s Scan Visually Explained - Algorithms.” *Dino Cajic - Chief Digital Officer & Author | Bridging IT, Marketing & Business*, 13 Dec. 2023, www.dinocajic.com/grahams-scan-visually-explained/.
5. CMU. *Fundamentals of Computational Geometry*, 22 Nov. 2022, www.cs.cmu.edu/~15451-f22/lectures/lec21-geometry.pdf.
6. Cook, Gerald, and Feitian Zhang. *Obstacle Mapping and Its Application to Robot Navigation | Part of Mobile Robots: Navigation, Control and Sensing, Surface Robots and Auvs | Wiley-IEEE Press Books | IEEE Xplore*, 2020, ieeexplore.ieee.org/document/8958876.
7. Cormen, Thomas H., et al. “Computational Geometry.” *Introduction to Algorithms*, Third ed., pp. 1014–1039.
8. gfg, vaibhav. “Convex Hull Algorithm.” *GeeksforGeeks*, GeeksforGeeks, 8 Aug. 2024, www.geeksforgeeks.org/convex-hull-algorithm/.
9. Goddard, Wayne. *Introduction to Algorithms Part 1: Divide and Conquer ...*, 2004, donaldlab.cs.duke.edu/Teaching/discretemath/Online/lectures/l23-convex-hull.pdf.
10. Goddard, Wayne. *Part A: Divide and Conquer; Sorting and Searching*, 2019, people.computing.clemson.edu/~goddard/texts/algor/AAA.pdf.
11. “Graham Scan.” *Graham Scan · Arcane Algorithm Archive*, www.algorithm-archive.org/contents/graham_scan/graham_scan.html. Accessed 30 Nov. 2024.
12. Graham, R. L. *An Efficient Algorithm for Determining the Convex Hull of a Finite Planar Set*, 28 Jan. 1972, mathweb.ucsd.edu/~ronspubs/72_10_convex_hull.pdf.
13. gurdevs. “Computational Geometry - Algorithms for Geometry.” *GeeksforGeeks*, GeeksforGeeks, 29 Jan. 2024, www.geeksforgeeks.org/what-is-computational-geometry-and-how-is-it-applied-in-solving-geometric-problems/.
14. Muthukrishnan. “Understanding Graham Scan Algorithm for Finding the Convex Hull of a Set of Points.” *Muthukrishnan*, 26 Jan. 2020, muthu.co/understanding-graham-scan-algorithm-for-finding-the-convex-hull-of-a-set-of-points/.
15. O’Connor, J J, and E F Robertson. “Ronald Graham - Biography.” *Maths History*, Feb. 2010, mathshistory.st-andrews.ac.uk/Biographies/Graham/.
16. Rathi, Aarti. “Convex Hull Using Graham Scan.” *GeeksforGeeks*, 8 Aug. 2024, www.geeksforgeeks.org/convex-hull-using-graham-scan/.
17. Sommer, Pascal. “A Gentle Introduction to the Convex Hull Problem.” *Medium*, Medium, 19 June 2020, medium.com/@pascal.sommer.ch/a-gentle-introduction-to-the-convex-hull-problem-62dfcabee90c.
18. Weisstein, Eric W. “Convex Hull.” *From Wolfram MathWorld*, mathworld.wolfram.com/ConvexHull.html. Accessed 30 Nov. 2024.
19. Weisstein, Eric W. “Polar Angle.” *From Wolfram MathWorld*, mathworld.wolfram.com/PolarAngle.html#:~:text=In%20the%20plane%2C%20the%20polar,the%20zenith%20angle%20and%20colatitude. Accessed 30 Nov. 2024.
20. Yadav, Madhav. “Graham Scan Algorithm.” *Medium*, Medium, 23 July 2023, medium.com/@madhavyadav4595/graham-scan-algorithm-5c287ef00ecd.