COT 6936: Topics in Algorithms

Giri Narasimhan

ECS 254A / EC 2474; Phone x3748; Email: giri@cs.fiu.edu
HOMEPAGE: http://www.cs.fiu.edu/~giri
https://moodle.cis.fiu.edu/v2.1/course/view.php?id=612

Apr 8, 2014

Presentation Outline

Applications

- Graphics

Applications

- Graphics
- Robotics

Applications

- Graphics
- Robotics
- Sensor Networks

Applications

- Graphics
- Robotics
- Sensor Networks
- ...

Presentation Outline

Convex Hull Problem

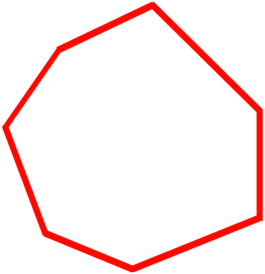
- This is a fundamental problem in Computational Geometry.
- There are many algorithms for solving this problem . . .

Convexity

- Convex Regions A region in space is called *convex* if line joining any two points in the region is completely contained in the region.

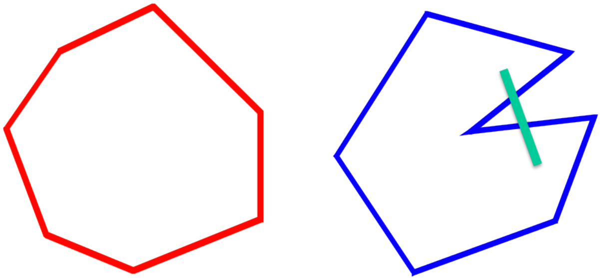
Convexity

- Convex Regions A region in space is called *convex* if line joining any two points in the region is completely contained in the region.



Convexity

- Convex Regions A region in space is called *convex* if line joining any two points in the region is completely contained in the region.

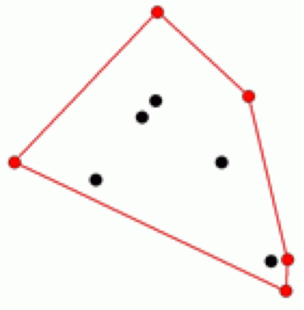


Convex Hulls

- Convex Hull: of a set of points, S, is the smallest convex region containing S.

Convex Hulls

- Convex Hull: of a set of points, S, is the smallest convex region containing S.

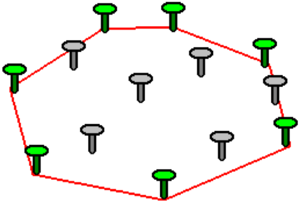


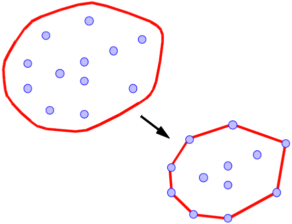
Convex Hulls – Rubber Band Analogy

- Convex Hull: of a set of points, S, is the smallest convex region containing S.

Convex Hulls – Rubber Band Analogy

- Convex Hull: of a set of points, S, is the smallest convex region containing S.





3-D Convex Hulls

- Convex Hull: of a set of points, S, is the smallest convex region containing S.

3-D Convex Hulls

- Convex Hull: of a set of points, S, is the smallest convex region containing S.

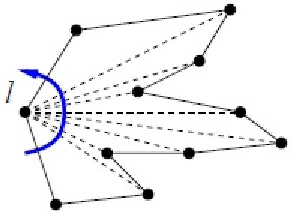
Presentation Outline

Convex Hulls

- Graham Scan algorithm:

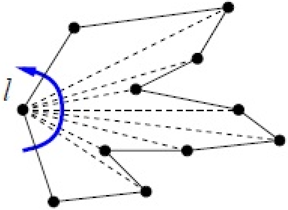
Convex Hulls

- Graham Scan algorithm:



Convex Hulls

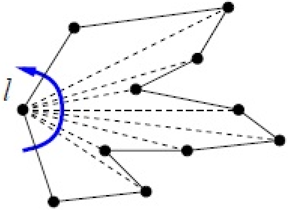
- Graham Scan algorithm:



`http://www.personal.kent.edu/~rmuhamma/Compgeometry/MyCG/ConvexHull/GrahamScan/grahamScan.htm`

Convex Hulls

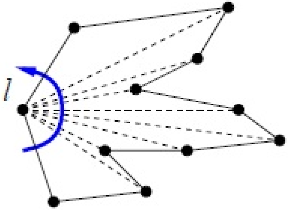
- Graham Scan algorithm:



http://www.personal.kent.edu/~rmuhamma/Compgeometry/MyCG/ConvexHull/GrahamScan/grahamScan.htm
Also shows proof of correctness;

Convex Hulls

- Graham Scan algorithm:



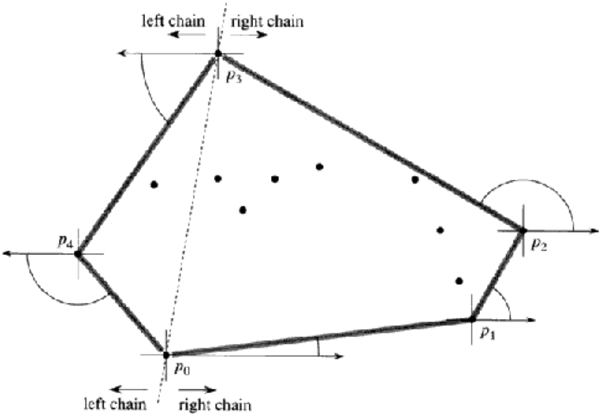
http://www.personal.kent.edu/~rmuhamma/Compgeometry/MyCG/ConvexHull/GrahamScan/grahamScan.htm
Also shows proof of correctness;
$O(n \log n)$ time, mainly for sorting

Convex Hulls

- Jarvis March algorithm:

Convex Hulls

- Jarvis March algorithm:



Package Wrapping – Jarvis March

- Time Complexity = (Cost of iteration) $\times$ (Number of iterations)

Package Wrapping – Jarvis March

- Time Complexity = (Cost of iteration) $\times$ (Number of iterations)
- Cost of each iteration

Package Wrapping – Jarvis March

- Time Complexity = (Cost of iteration) $\times$ (Number of iterations)
- Cost of each iteration = $O(n)$

Package Wrapping – Jarvis March

- Time Complexity $=$ (Cost of iteration) $\times$ (Number of iterations)
- Cost of each iteration $= O(n)$
- Number of iterations

Package Wrapping – Jarvis March

- Time Complexity $=$ (Cost of iteration) $\times$ (Number of iterations)
- Cost of each iteration $= O(n)$
- Number of iterations $= O(n)$

Package Wrapping – Jarvis March

- Time Complexity = (Cost of iteration) $\times$ (Number of iterations)
- Cost of each iteration = $O(n)$
- Number of iterations = $O(n)$
  - It is really $O(h)$, where h = number of points on hull

Package Wrapping – Jarvis March

- Time Complexity = (Cost of iteration) $\times$ (Number of iterations)
- Cost of each iteration = $O(n)$
- Number of iterations = $O(n)$
  - It is really $O(h)$, where h = number of points on hull
- Total cost = $O(nh)$

Package Wrapping – Jarvis March

- Time Complexity = (Cost of iteration) $\times$ (Number of iterations)
- Cost of each iteration $= O(n)$
- Number of iterations $= O(n)$
  - It is really $O(h)$, where $h =$ number of points on hull
- Total cost $= O(nh)$ [Output-Sensitive analysis]

Package Wrapping – Jarvis March

- Time Complexity = (Cost of iteration) $\times$ (Number of iterations)
- Cost of each iteration = $O(n)$
- Number of iterations = $O(n)$
  - It is really $O(h)$, where h = number of points on hull
- Total cost = $O(nh)$ [Output-Sensitive analysis]
- Which is better $O(n \log n)$ or $O(nh)$?

Package Wrapping – Jarvis March

- Time Complexity = (Cost of iteration) $\times$ (Number of iterations)
- Cost of each iteration = $O(n)$
- Number of iterations = $O(n)$
  - It is really $O(h)$, where h = number of points on hull
- Total cost = $O(nh)$ [Output-Sensitive analysis]
- Which is better $O(n \log n)$ or $O(nh)$?
- Lower bound

Package Wrapping – Jarvis March

- Time Complexity = (Cost of iteration) × (Number of iterations)
- Cost of each iteration $= O(n)$
- Number of iterations $= O(n)$
  - It is really $O(h)$, where $h =$ number of points on hull
- Total cost $= O(nh)$ [Output-Sensitive analysis]
- Which is better $O(n \log n)$ or $O(nh)$?
- Lower bound $= \omega(n \log h)$

Presentation Outline

Chan's Algorithm

- Combines the benefits of both algorithms

Chan's Algorithm

- Combines the benefits of both algorithms
- Partition points into n/m groups of size m

Chan's Algorithm

- Combines the benefits of both algorithms
- Partition points into n/m groups of size m
- Use Graham Scan on each group.

Chan's Algorithm

- Combines the benefits of both algorithms
- Partition points into n/m groups of size m
- Use Graham Scan on each group.
- Total time $= O((m \log m) \cdot (n/m)) = O(n \log m)$

Chan's Algorithm

- Combines the benefits of both algorithms
- Partition points into n/m groups of size m
- Use Graham Scan on each group.
- Total time $= O((m \log m) \cdot (n/m)) = O(n \log m)$
- Merge the n/m convex hulls using the Jarvis Macrh algorithm by treating each group as a big point.

Chan's Algorithm

- Combines the benefits of both algorithms
- Partition points into n/m groups of size m
- Use Graham Scan on each group.
- Total time $= O((m \log m) \cdot (n/m)) = O(n \log m)$
- Merge the n/m convex hulls using the Jarvis Macrh algorithm by treating each group as a big point.
  - Tangent between point and convex hull with m points can be computed in $O(\log m)$ time.

Chan's Algorithm

- Combines the benefits of both algorithms
- Partition points into n/m groups of size m
- Use Graham Scan on each group.
- Total time $= O((m \log m) \cdot (n/m)) = O(n \log m)$
- Merge the n/m convex hulls using the Jarvis Macrh algorithm by treating each group as a big point.
  - Tangent between point and convex hull with m points can be computed in $O(\log m)$ time.
  - Total time $=$

Chan's Algorithm

- Combines the benefits of both algorithms
- Partition points into n/m groups of size m
- Use Graham Scan on each group.
- Total time $= O((m \log m) \cdot (n/m)) = O(n \log m)$
- Merge the n/m convex hulls using the Jarvis Macrh algorithm by treating each group as a big point.
  - Tangent between point and convex hull with m points can be computed in $O(\log m)$ time.
  - Total time $=$

$$O((n/m)(\log m)(h))$$

Chan's Algorithm

- Combines the benefits of both algorithms
- Partition points into n/m groups of size m
- Use Graham Scan on each group.
- Total time $= O((m \log m) \cdot (n/m)) = O(n \log m)$
- Merge the n/m convex hulls using the Jarvis Macrh algorithm by treating each group as a big point.
  - Tangent between point and convex hull with m points can be computed in $O(\log m)$ time.
  - Total time $=$

$$O((n/m)(\log m)(h)) = O((n/m)h \log m)$$

Chan's Algorithm

- Time Complexity

Chan's Algorithm

- Time Complexity $= O(n \log m + \frac{n}{m} h \log m)$

Chan's Algorithm

- Time Complexity $= O(n \log m + \frac{n}{m} h \log m)$
- If $m = h$, then Time

Chan's Algorithm

- Time Complexity $= O(n \log m + \frac{n}{m} h \log m)$
- If $m = h$, then Time $= O(n \log h)$

Chan's Algorithm

- Time Complexity $= O(n \log m + \frac{n}{m} h \log m)$
- If $m = h$, then Time $= O(n \log h)$
- Problem: We don't know h!

Chan's Algorithm

- Time Complexity $= O(n \log m + \frac{n}{m} h \log m)$
- If $m = h$, then Time $= O(n \log h)$
- Problem: We don't know h!
- Guess h . . .

Chan's Algorithm

- Time Complexity $= O(n \log m + \frac{n}{m} h \log m)$
- If $m = h$, then Time $= O(n \log h)$
- Problem: We don't know h!
- Guess h ... How?

Chan's Algorithm

- Time Complexity $= O(n \log m + \frac{n}{m} h \log m)$
- If $m = h$, then Time $= O(n \log h)$
- Problem: We don't know h!
- Guess h ... How?
  - Linear Search: $O(nh \log h)$

Chan's Algorithm

- Time Complexity $= O(n \log m + \frac{n}{m} h \log m)$
- If $m = h$, then Time $= O(n \log h)$
- Problem: We don't know h!
- Guess h ... How?
    - Linear Search: $O(nh \log h)$
    - Binary Search: $O(n \log^2 h)$

Chan's Algorithm

- Time Complexity $= O(n \log m + \frac{n}{m} h \log m)$
- If $m = h$, then Time $= O(n \log h)$
- Problem: We don't know h!
- Guess h . . . How?
  - Linear Search: $O(nh \log h)$
  - Binary Search: $O(n \log^2 h)$
  - Doubling Search:

Chan's Algorithm

- Time Complexity $= O(n \log m + \frac{n}{m} h \log m)$
- If $m = h$, then Time $= O(n \log h)$
- Problem: We don't know h!
- Guess h ... How?
  - Linear Search: $O(nh \log h)$
  - Binary Search: $O(n \log^2 h)$
  - Doubling Search: Try $m = 1, 2, 4, 8, \ldots$

Chan's Algorithm

- Time Complexity $= O(n \log m + \frac{n}{m} h \log m)$
- If $m = h$, then Time $= O(n \log h)$
- Problem: We don't know h!
- Guess h . . . How?
  - Linear Search: $O(nh \log h)$
  - Binary Search: $O(n \log^2 h)$
  - Doubling Search: Try $m = 1, 2, 4, 8, \ldots$
    Time Complexity $= O(n \log^2 h)$

Finishing the Analysis

- Another idea:

Finishing the Analysis

- Another idea: What if $m = h^2$?

Finishing the Analysis

- Another idea: What if $m = h^2$?
- Then $O(n \log m) = O(n \log h)$

Finishing the Analysis

- Another idea: What if $m = h^2$?
- Then $O(n \log m) = O(n \log h)$
- Try $m = 2, 4, 16, 256, \ldots$.

Finishing the Analysis

- Another idea: What if $m = h^2$?
- Then $O(n \log m) = O(n \log h)$
- Try $m = 2, 4, 16, 256, \ldots$.
- Analysis:

Finishing the Analysis

- Another idea: What if $m = h^2$?
- Then $O(n \log m) = O(n \log h)$
- Try $m = 2, 4, 16, 256, \ldots$.
- Analysis:

$$\sum_{t=1}^{\lg \lg h} n2^t = n \sum_{t=1}^{\lg \lg h} 2^t \leq n2^{1+\lg \lg h} = 2n \lg h = O(n \log h)$$