Intro to course

web page https://cs.uwaterloo.ca/~alubiw/CS763.html
Piazza https://piazza.com/uwaterloo.ca/fall2020/cs763/home

Credit:

- 6 assignments (roughly 2 questions each) (50%)
- a project (50%). Pick some topic that interests you and is relevant to the course; explore some aspect of it. You may attempt original research or report on some papers (one paper deeply or a few papers less deeply). You must do a written report and a class presentation. I will suggest possible topics.

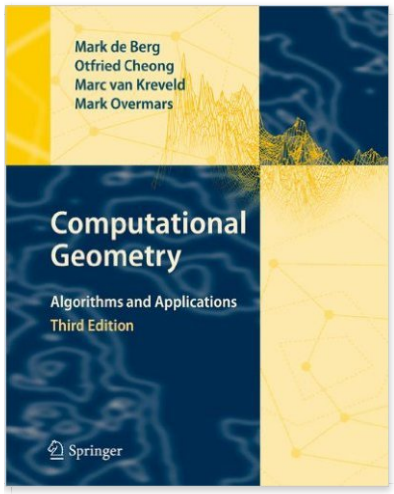
Course Outline

- polygon triangulation
- visibility and guarding
- convex hulls
- linear programming
- Voronoi diagrams and Delaunay triangulations
- surface reconstruction
- arrangements and duality
- geometric data structures, search problems
- motion planning, shortest paths
- curves, trajectories, Fréchet distance

Background: I will assume a background in algorithms and data structures from a decent undergraduate course (e.g. UW's CS 341).

Resources

text:
Computational Geometry: Algorithms and Applications
[CGAA] “the 3 Marks book”



https://ocul-wtl.primo.exlibrisgroup.com/permalink/01OCUL_WTL/156lh75/cdi_springer_primary_978-3-540-77974-2_45973

lecture notes:
Geometry: Combinatorics & Algorithms
[Zurich notes]

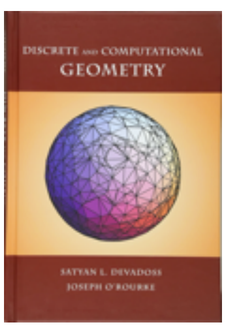
https://geometry.inf.ethz.ch/gca18.pdf

to find papers

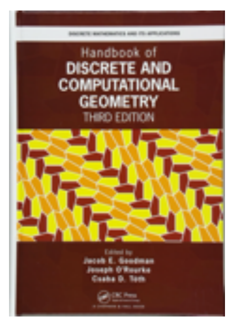
https://scholar.google.com

other good books

Discrete and Computational Geometry
[Devadoss-O’Rourke]

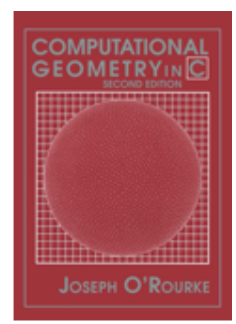


Handbook of Discrete and
Computational Geometry
[Handbook]

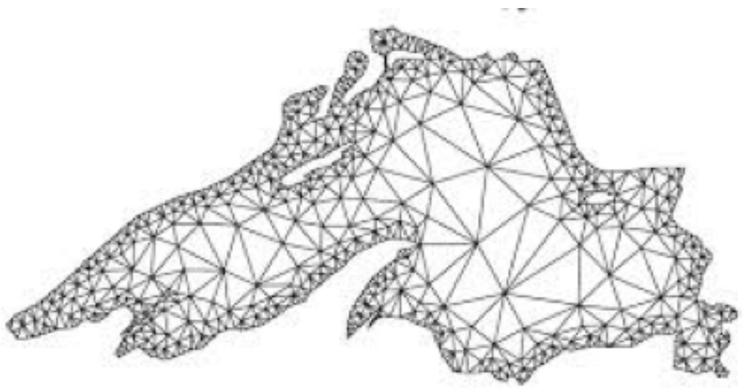


https://ocul-wtl.primo.exlibrisgroup.com/permalink/01OCUL_WTL/156lh75/cdi_askewsholts_vlebooks_9781498711425

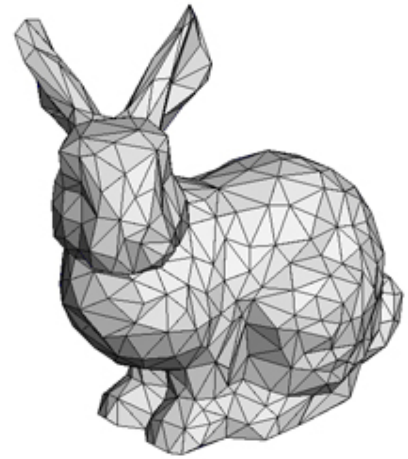
Computational Geometry in C
[O’Rourke]



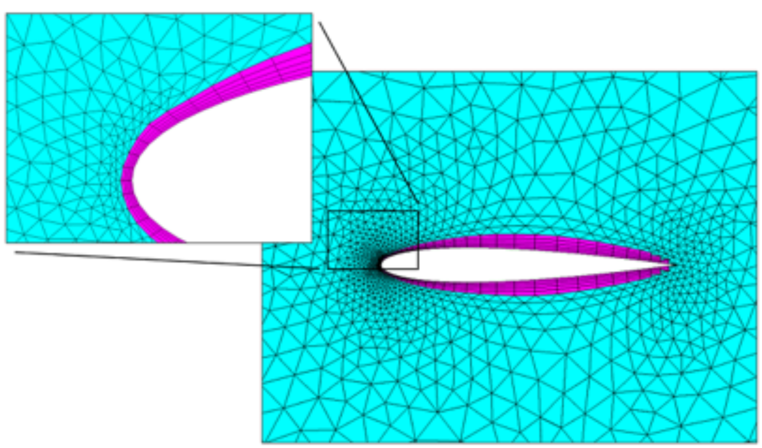
Triangulate a polygon/point set/surface



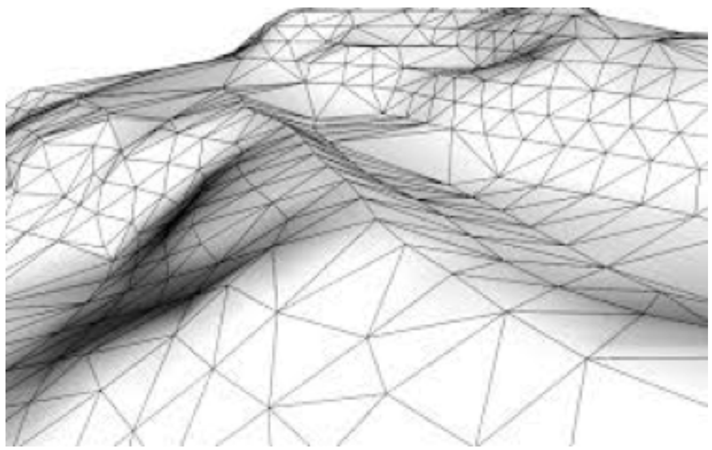
polygon



polyhedron



polygonal region



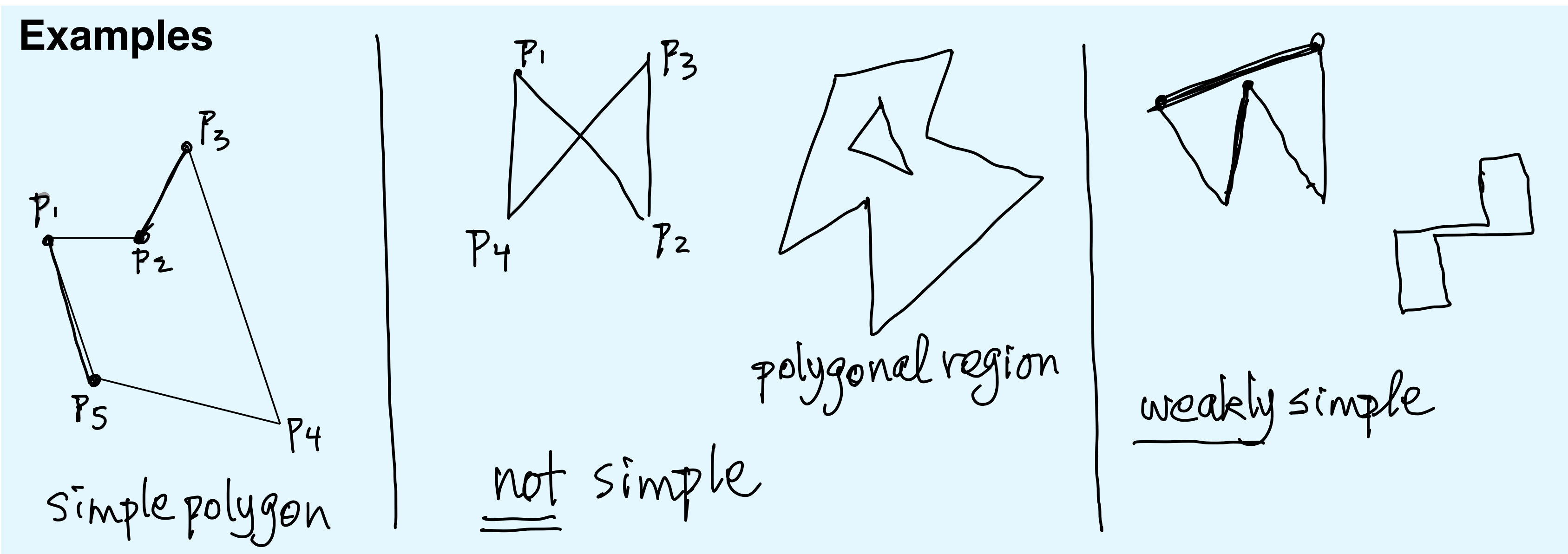
terrain

Polygon Triangulation

Definition. A ***polygon*** is specified by a sequence of points in the plane,

$p_1, p_2, \ldots, p_n$ called ***vertices***. The ***edges*** are the line segments $e_i = p_i p_{i+1}$

We assume ***simple*** polygons — two edges intersect only at a common vertex.



How do we test if a polygon is simple? Plane-sweep $O(n \log n)$.

. . . weakly simple?

Recognizing weakly simple polygons
HA Akitaya, G Aloupis, J Erickson, CD Tóth - Discrete & Computational …, 2017 - Springer
https://link.springer.com/article/10.1007/s00454-017-9918-3

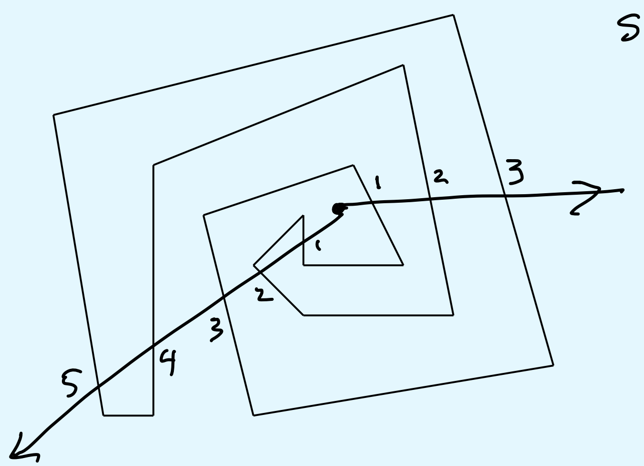
Jordan Curve Theorem

A simple polygon divides the plane into two regions, the ***inside*** and the ***outside***.

True more generally for simple curves. https://en.wikipedia.org/wiki/Jordan_curve_theorem

Elementary proof for polygons — Courant and Robbins, 1941
https://en.wikipedia.org/wiki/What_Is_Mathematics%3F

How to test if a point is inside/outside a polygon:



shoot ray & count
\# intersection
odd ⇒ point inside

Motivation for Decomposing Polygons

Most algorithms on polygons work better on small/nice polygons — triangles or convex pieces.

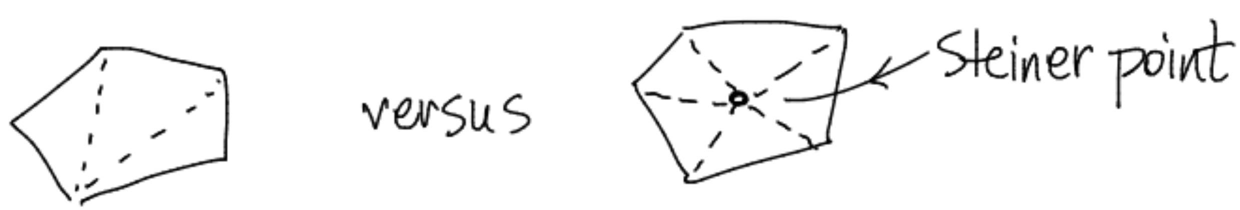
Note: 3D is more useful than 2D but we often work with *surfaces* in 3D and these are stored as a collection of polygons.

Types of Decompositions

- partition — express polygon as union of disjoint subpolygons
- covering — express polygon as union of subpolygons
- Boolean combination — express polygon as Boolean combination (union, intersection, minus, etc.) of subpolygons.

Steiner points

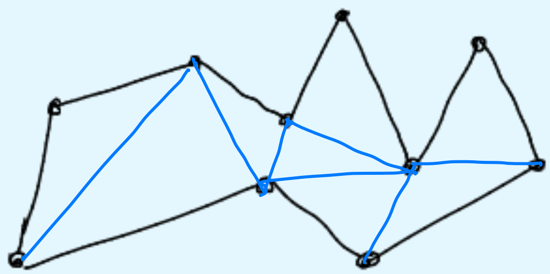
Sometimes we require the subpolygon vertices to be vertices of the original.
Otherwise the new vertices are called ***Steiner points***.
Examples:



Triangulating Polygons

Partition a polygon into triangles without Steiner points. Each triangle edge will be a ***chord*** — a line segment inside the polygon joining two vertices.

Example



Theorem [Lennes 1911] Any polygon can be triangulated.

Proof. By induction.

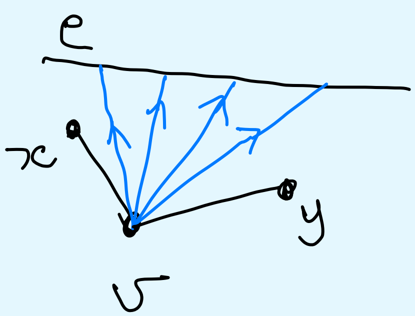
Enough to find one chord. Divides polygon into smaller polygons

Then done by induction.

Basis. $n=3$

Theorem [Lennes 1911] Any polygon can be triangulated.

Proof. Consider polygon with $n \geq 4$. Consider vertex v



v adjacent to vertices x, y

If $\exists$ chord (v, z) done

— v "sees" z.

Otherwise all rays from v hit edge e (always the same edge)

$\therefore$ angle at v is convex

and (x, y) is inside polygon

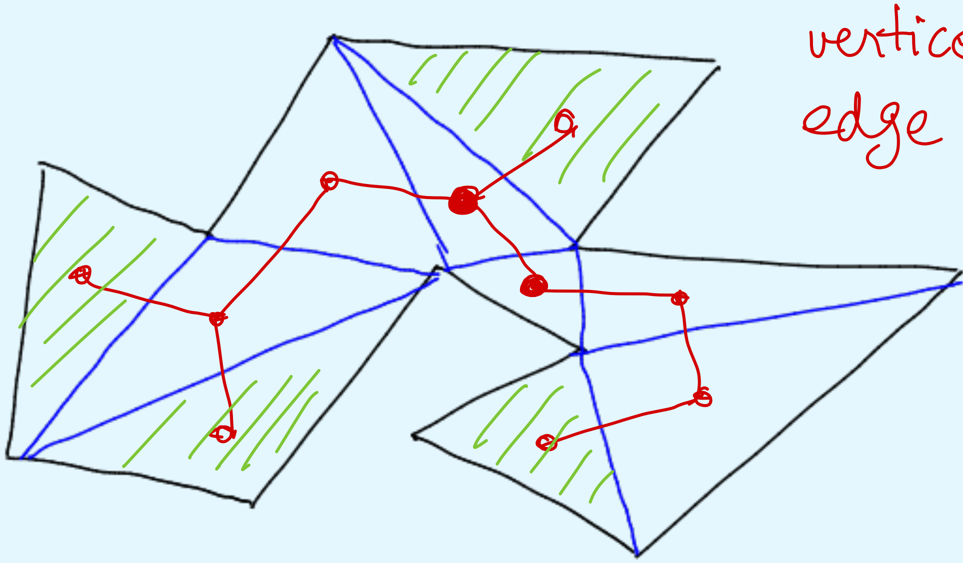
e may be incident to x or y but not both.

$\therefore$ (x, y) is a chord. $\square$

Some properties of polygon triangulations

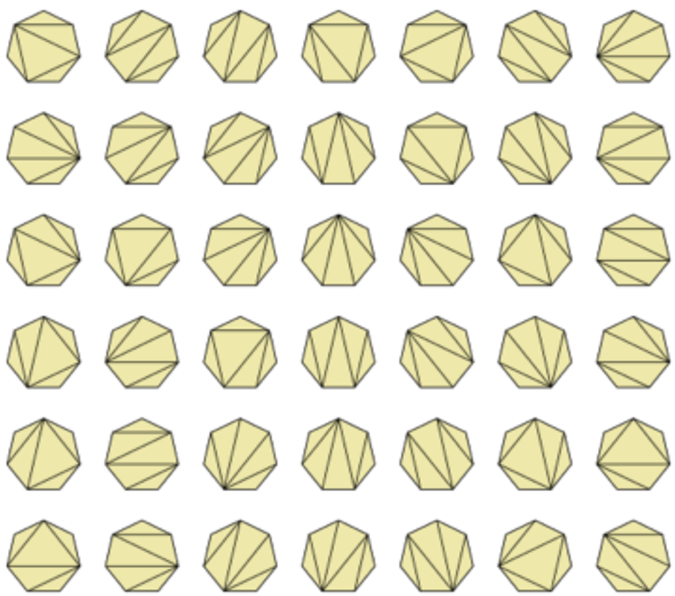
- number of triangles is $n-2$ # chords $n-3$
- every polygon has (at least) two disjoint ***ears*** – 2 polygon edges + chord forming a triangle
- triangles form a tree – Dual

vertices = triangles
edge = adjacent triangles



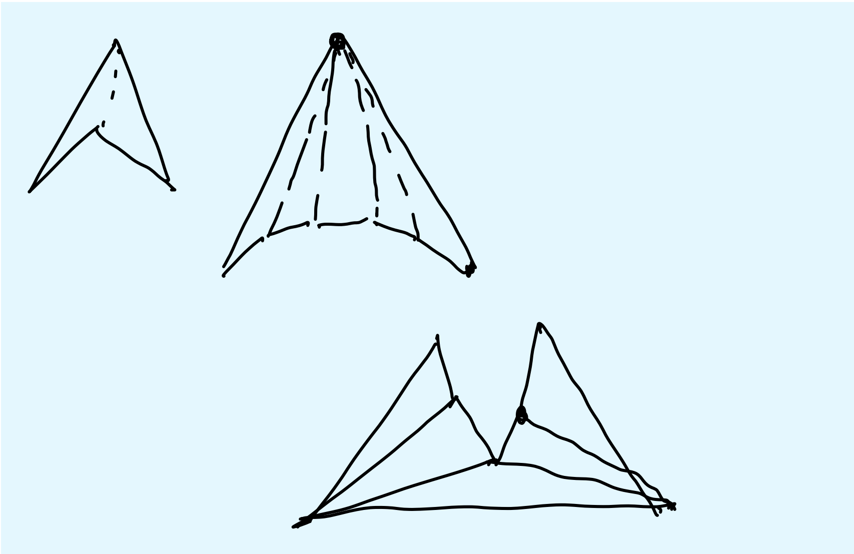
Exercise. Prove that the sum of the interior angles of any polygon is $\pi(n-2)$

The number of triangulations of a polygon.



The 42 possible triangulations for a convex heptagon
https://en.wikipedia.org/wiki/Polygon_triangulation

Some polygons have a unique triangulation.



Fact: The number of triangulations of an n-vertex convex polygon is the Catalan number C_{n-2}

Problem: Give a polynomial-time algorithm to compute the number of triangulations of a simple polygon.

for point sets

Peeling and Nibbling the Cactus: Subexponential-Time Algorithms for Counting Triangulations and Related Problems

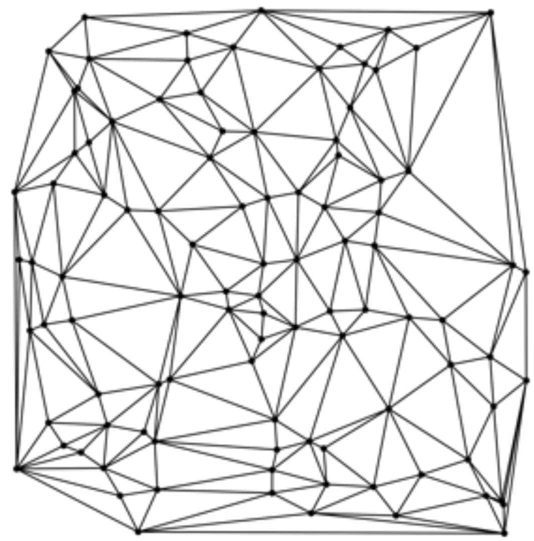
D Marx, T Miltzow - 32nd International Symposium on …, 2016 - drops.dagstuhl.de
https://drops.dagstuhl.de/opus/volltexte/2016/5944/

for polygonal regions

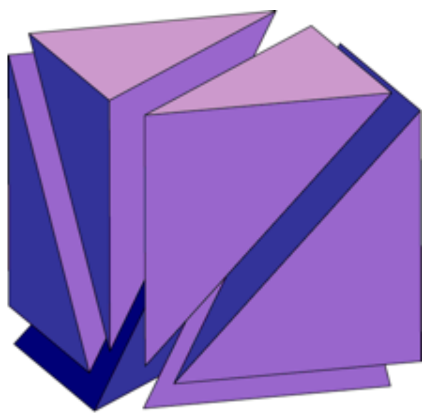
Counting Polygon Triangulations is Hard

D Eppstein
https://arxiv.org/pdf/1903.04737

Later on we will talk about triangulating point sets in the plane



Also about “triangulating” polyhedra in 3D



Exercise: cut a cube into min. number of tetrahedra.

Algorithms to triangulate a polygon

1. obvious method (find a chord, following the proof) takes O(n^4)

   can be improved to O(n^2) by cutting off ears

   Exercise: figure out the details for this

2. O(n log n) algorithm next day

3. O(n log* n) randomized algorithm of Seidel (faster than O(n log n))

4. Optimal algorithm to triangulate a polygon O(*n*)

Bernard Chazelle 1991

[Triangulating a simple polygon in linear time](http://link.springer.com/article/10.1007/BF02574703) http://link.springer.com/article/10.1007/BF02574703

B Chazelle - Discrete & Computational Geometry, 1991 - Springer

Abstract. We give a deterministic algorithm for triangulating a simple polygon in linear time. The basic strategy is to build a coarse approximation of a triangulation in a bottom-up phase and then use the information computed along the way to refine the triangulation ...

But so complicated that there's no implementation!

> Linear-time polygon triangulation has intriguing consequences. For example, one cannot check in linear time whether a list of segments ab, cd, ef, gh, etc, is free of intersections, but if the list is of the form ab, bc, cd, de, etc, then miraculously one can. Segueing into my favorite open problem in plane geometry, can the self-intersections of a polygonal curve be computed in linear time? I know the answer (it's yes) but not the proof.
>
> https://www.cs.princeton.edu/~chazelle/linernotes.html

$O(n+k)$

intersections

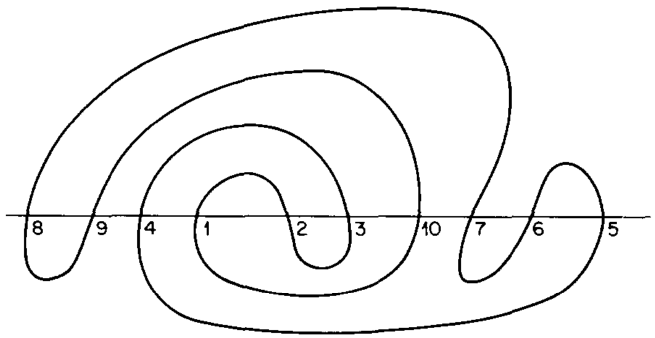
The power of having a simple polygon

https://www.sciencedirect.com/science/article/pii/002001909090111A

Simplified linear-time Jordan sorting and polygon clipping

KY Fung, TM Nicholl, RE Tarjan, CJ Van Wyk - Information Processing …, 1990 - Elsevier

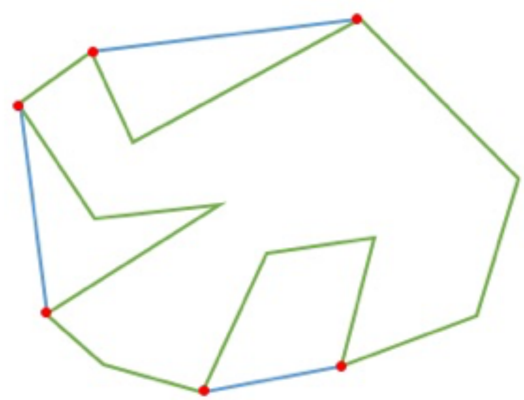
Given the intersection points of a Jordan curve with the x-axis in the order in which they occur along the curve, the Jordan sorting problem is to sort them into the order in which they occur along the x-axis. This problem arises in clipping a simple polygon against a rectangle (a “window”) and in efficient algorithms for triangulating a simple polygon. Hoffman, Mehlhorn, Rosenstiehl, and Tarjan proposed an algorithm that solves the Jordan sorting problem in time that is linear in the number of intersection points, but their algorithm requires …



ON-LINE CONSTRUCTION OF THE CONVEX HULL OF A SIMPLE POLYLINE

AA Melkman - Information Processing Letters, 1987 - ime.usp.br

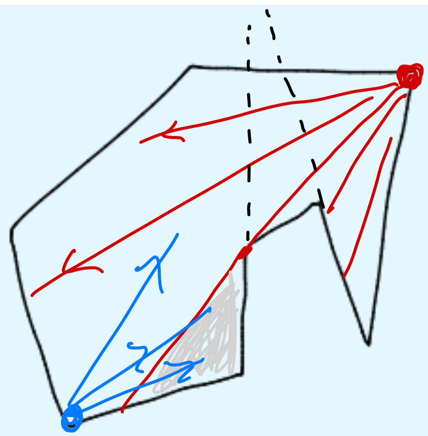
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.512.9681&rep=rep1&type=pdf



Art Gallery Theorem (an application of triangulations)

Regard a polygon as a floorplan of an art gallery, edges = walls.
How many guards are needed to watch the whole gallery?

Example



2 guards can guard this polygon

1 guard cannot.

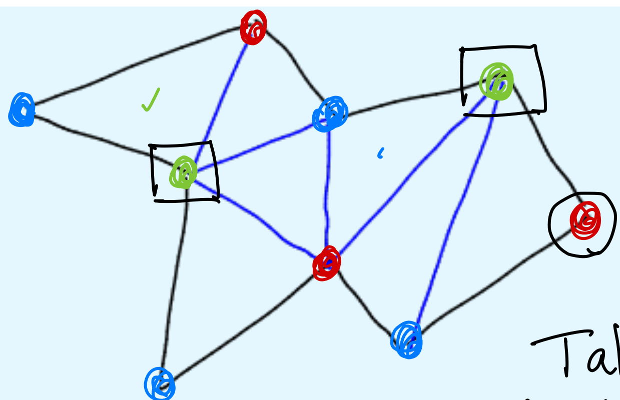
Problem posed by Victor Klee 1973, bound proved by Chvatal 1975, simple proof by Fisk 1978.

Theorem. For an n vertex polygon floor(n/3) guards always suffice, and for some n-vertex polygons, floor(n/3) guards are necessary.

Theorem. For an *n* vertex polygon floor(*n*/3) guards always suffice, and for some *n*-vertex polygons, floor(*n*/3) guards are necessary.

Proof [Fisk] Triangulate the polygon.

Colour the vertices red, green, blue s.t. every triangle has every colour.

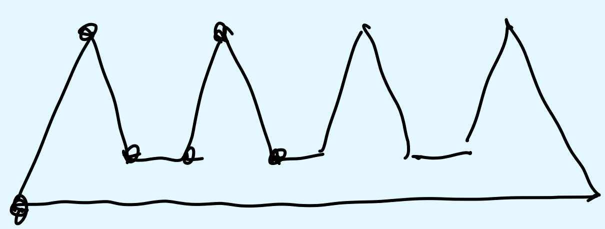


pf. cut off one ear. By induction, colour smaller polygon. Then colour tip of ear with required colour.

Take smallest colour class. # pts is $\leq \lfloor \frac{n}{3} \rfloor$ This set guards all triangles ∴ whole polygon.

Note: For convex, or "star shaped" one guard works.

When are floor(*n* /3) guards necessary?

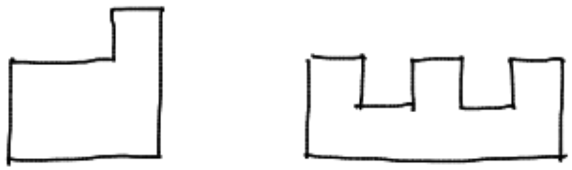


$n=12$ need $\frac{n}{3}=4$ guards

Need one guard per "tooth"

There are many further results on guarding.

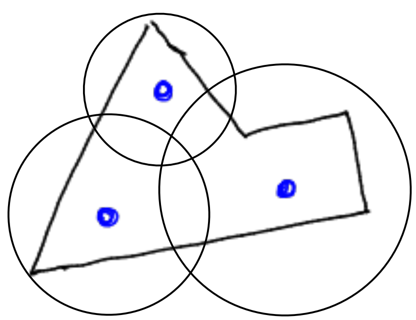
Exercise: If the polygon is orthogonal, make a conjecture about the number of guards that are always sufficient and sometimes necessary.



Art Gallery Theorems and Algorithms, Joseph O’Rourke, 1987

http://cs.smith.edu/~jorourke/books/ArtGalleryTheorems/

Guards may have limited visibility — sensor networks



Efficient sensor placement for surveillance problems

PK Agarwal, E Ezra, SK Ganjugunte - International Conference on …, 2009 - Springer

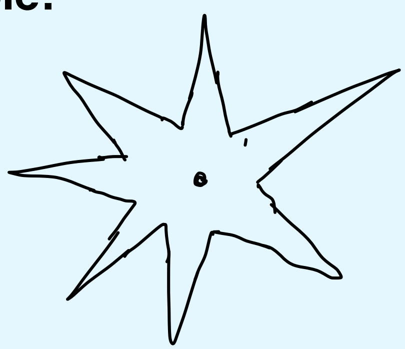
https://link.springer.com/chapter/10.1007/978-3-642-02085-8_22

Algorithms for the Art Gallery Problem

Is there an algorithm to find the minimum number of guards for a given polygon?
(Above results were about the worst-case number of guards for an n-gon.)

Guards need not be on the boundary.

Example:

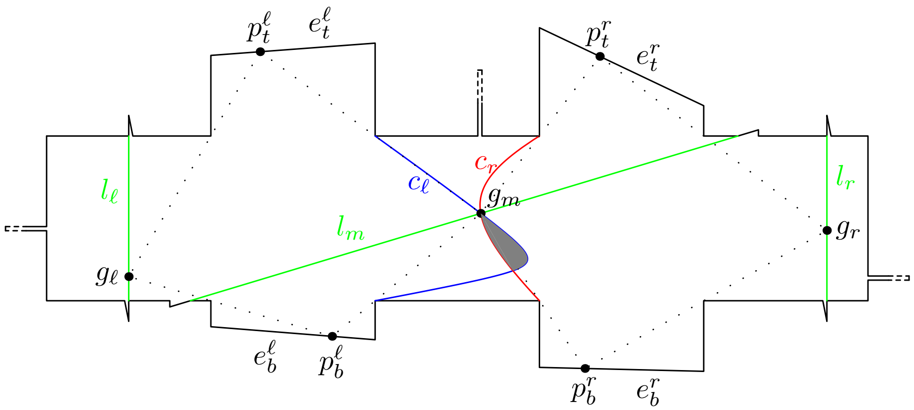


one guard inside is enough
but not on vertices.
if the guard must be

The problem is NP-hard. Is the decision problem in NP?

Geometric problems involve issues of real numbers!
What is our model of computing? How do we deal with imprecise points?

Guards might need to be at irrational points! (to get min. number of guards)



https://drops.dagstuhl.de/opus/volltexte/2017/7194/

Irrational Guards are Sometimes Needed

M Abrahamsen, A Adamaszek… - … Geometry (SoCG 2017), 2017 - drops.dagstuhl.de

The Art Gallery Problem is hard for existential theory of the reals $\exists\mathbb{R}$

The art gallery problem is∃ ℝ-complete

M Abrahamsen, A Adamaszek, T Miltzow - Proceedings of the 50th …, 2018 - dl.acm.org

https://dl.acm.org/doi/10.1145/3188745.3188868

$$P \subseteq NP \subseteq \exists\mathbb{R} \subseteq PSPACE$$

A Practical Algorithm with Performance Guarantees for the Art~ Gallery Problem

S Hengeveld, T Miltzow - arXiv preprint arXiv:2007.06920, 2020 - arxiv.org

https://arxiv.org/abs/2007.06920

Summary

- polygon, triangulation, art gallery problem
- two proofs: polygons can be triangulated; n/3 art gallery guards
- dangers of real numbers in geometric problems
- algorithms! possible, impossible, un-implementable

References

- [CGAA] Section 3.1
- [Zurich notes] Chapter 3
- [O'Rourke] 1.1, 1.2
- [Devadoss-O'Rourke] 1.1 - 1.3