

CS 164 & CS 266: Computational Geometry

Lecture 0: Introduction; Area and Coordinates

David Eppstein

University of California, Irvine

Fall Quarter, 2026



This work is licensed under a Creative Commons Attribution 4.0 International License

About the course

General Course Information

Instructor: David Eppstein

Teaching assistant: Alvin Chiu

Resources: Course info: <https://www.ics.uci.edu/~eppstein/164>

Ed Discussion (online forum for course-related questions)

Gradescope (for exam scores)

Coursework

Weekly problem sets (not graded!), two midterms, and final exam

We will provide problem set solutions (a week later)

Exams are not cumulative

(not intended to test you on earlier material but may assume it as background)

They will be short-answer, in-person, closed book and closed notes.

For CS266 students: Weekly reading: a research paper related to each week's topics

You should understand what the paper is claiming to do (and may be tested on this)

Undergrads are welcome to read these too! (But you will not be tested on them.)

Why are the problem sets not graded?

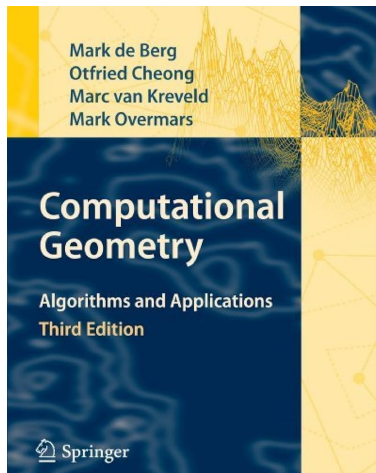
Main reason: I don't want to give an unfair advantage to students who look up the answers (or ask an LLM), compared to the students who do their own work

Secondary reasons:

- ▶ Similar questions will be on the exams (or in some cases the same), so having them in problem sets (rather than keeping homework and exam questions separate) should provide more opportunity for practice
- ▶ I can assign problems that match each week's lectures, rather than delaying them until those lectures are complete

Takeaway message: you should set aside time to work on the problems because it will show up in your exam scores

Textbook



Computational Geometry: An Introduction (3rd ed.)

Mark de Berg, Otfried (“Mark”) Cheong, Marc van Kreveld, Mark Overmars

Strongly recommended and free online from UCI internet addresses via a campus subscription to Springer books:

<https://link.springer.com/book/10.1007/978-3-540-77974-2>

Syllabus provides readings corresponding to lecture content

Area

A toy example

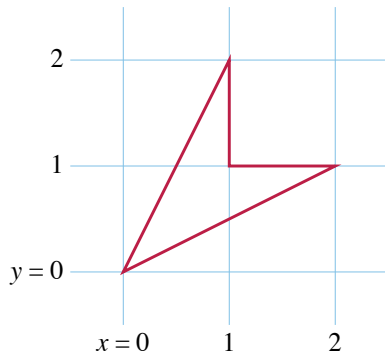
What is the area of this polygon?

Clever solution:

Cut along the grid lines

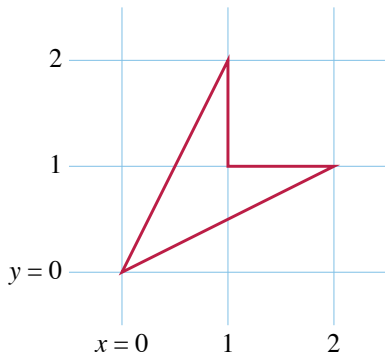
Flip the cut-off triangles into the lower left kite

The result exactly covers a square, area 1



Area of polygons

We want a solution without cleverness that a computer can follow



Vertex coordinates:

$[(0, 0), (1, 2), (1, 1), (2, 1)]$

Area: 1

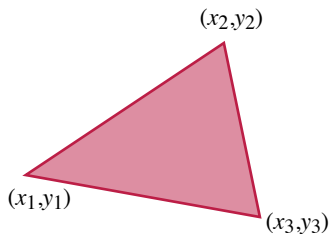
Input: Clockwise sequence of vertices

Each given by integer Cartesian coordinates

Output: A number, the area

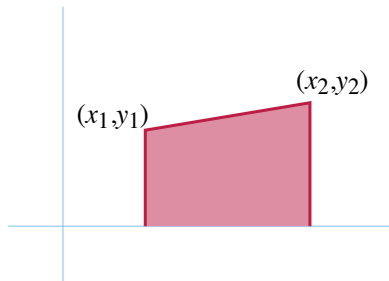
Idea: decompose into simpler shapes

Triangles



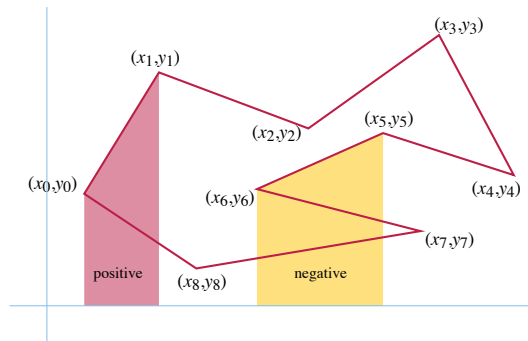
$$A = \frac{1}{2} \det \begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{pmatrix}$$
$$= \frac{1}{2} (x_1 y_2 - x_1 y_3 + x_2 y_3 - x_2 y_1 + x_3 y_1 - x_3 y_2)$$

Trapezoids



$$A = \frac{1}{2} (x_2 - x_1) (y_1 + y_2)$$

Using trapezoids to compute area



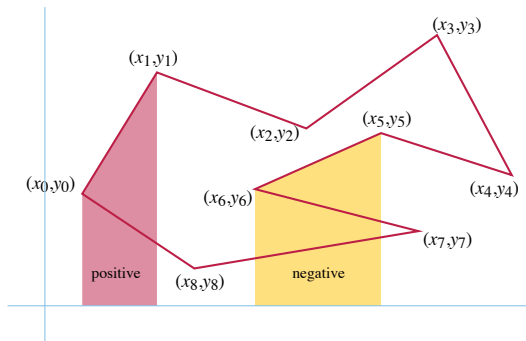
Add the areas of trapezoids below each upper edge of the polygon

Subtract the areas of trapezoids below each lower edge

Outside polygon, positive and negative areas cancel

Inside points all covered by one more positive than negative

Trapezoid formula for area



$$A = \sum_{i=0}^{n-1} \frac{1}{2} (x_{i+1} - x_i)(y_i + y_{i+1})$$

All indexes computed modulo n so $(x_n, y_n) = (x_0, y_0)$

Produces a positive number for positive trapezoids,
negative for negative trapezoids

Trapezoid formula as an algorithm

Summation $\Rightarrow$ for-loop

```
def area(P):  
    area = 0  
    for i in 0, 1, 2, ... n-1:  
        j = (i + 1) mod n  
        xi,yi = P[i]  
        xj,yj = P[j]  
        area += (xj-xi)*(yj+yi)/2  
    return area
```

Easy, time = $O(n)$

Coordinate systems

Coordinate systems for points in the plane

Cartesian coordinates (x, y)

Simple, familiar

Generalize to higher dims

Widely used, familiar

Polar coordinates (r, θ)

Angle and distance from origin

Widely known, not as useful

Complex numbers

Built into some programming languages (Python)

Make certain transformations easy:

Transformation	Formula
translate by t	$q \mapsto q + t$
scale by s	$q \mapsto qs$
rotate by θ	$q \mapsto q(\cos \theta + i \sin \theta)$

Not as easy to generalize to higher dimensions

Formulas for coordinate conversion

Cartesian to polar

$$r = \sqrt{x^2 + y^2} = \text{hypot}(x, y)$$

$$\theta = \text{atan2}(y, x)$$

Polar to Cartesian

$$x = r \cdot \cos \theta$$

$$y = r \cdot \sin \theta$$

Cartesian to complex

$$q = x + i y = x + 1j * y$$

Complex to Cartesian

$$x = \Re(q) = q.\text{real}$$

$$y = \Im(q) = q.\text{imag}$$