Computational Geometry

Michael T. Goodrich

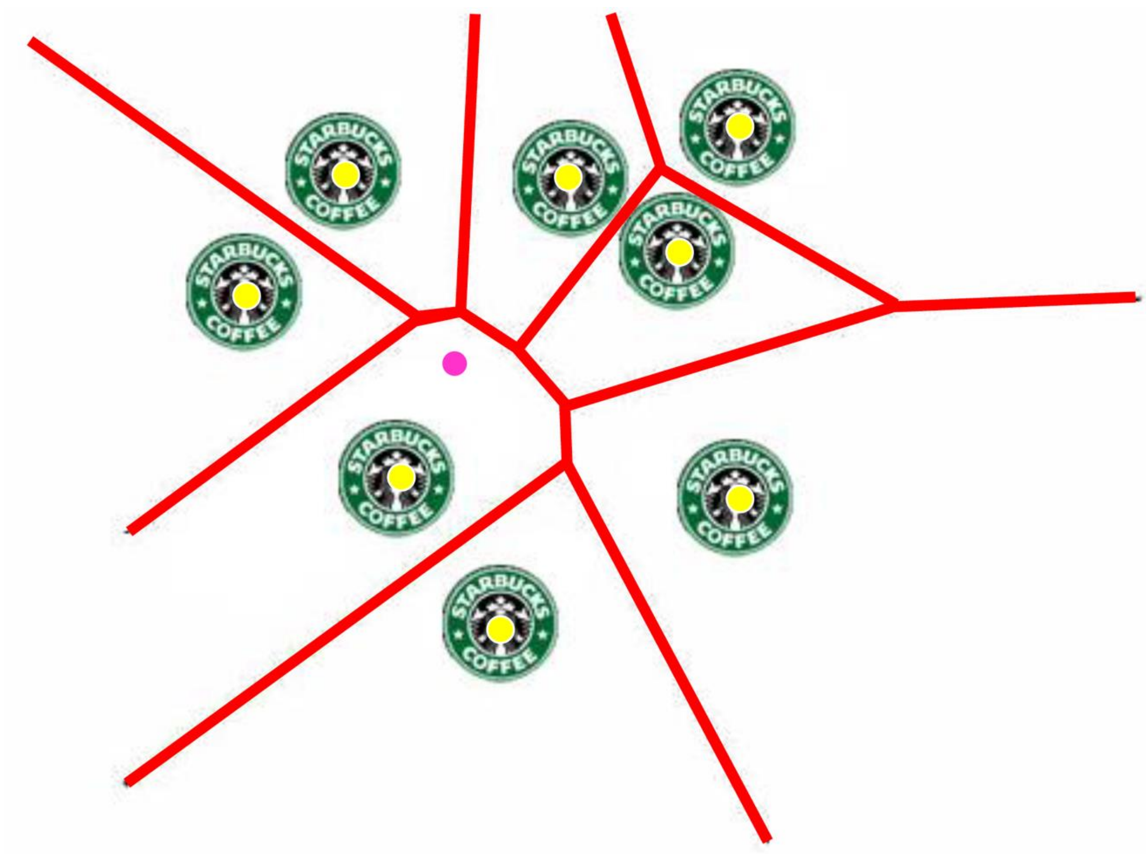
Introduction

Convex Hulls

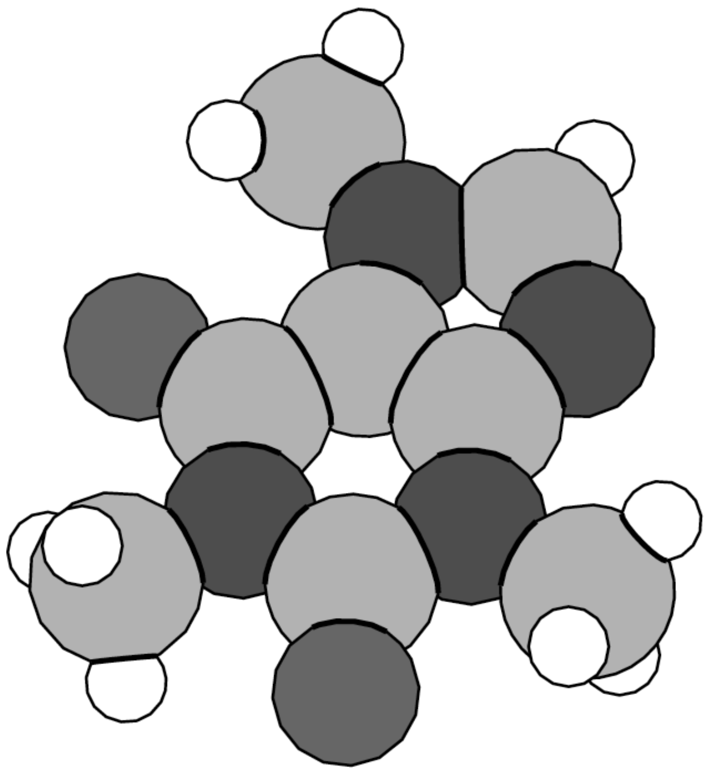
Definition

- **Computational geometry** involves the design, analysis and implementation of efficient algorithms for solving geometric problems, e.g., problems involving points, lines, segments, triangles, polygons, etc.

Application: Location Data

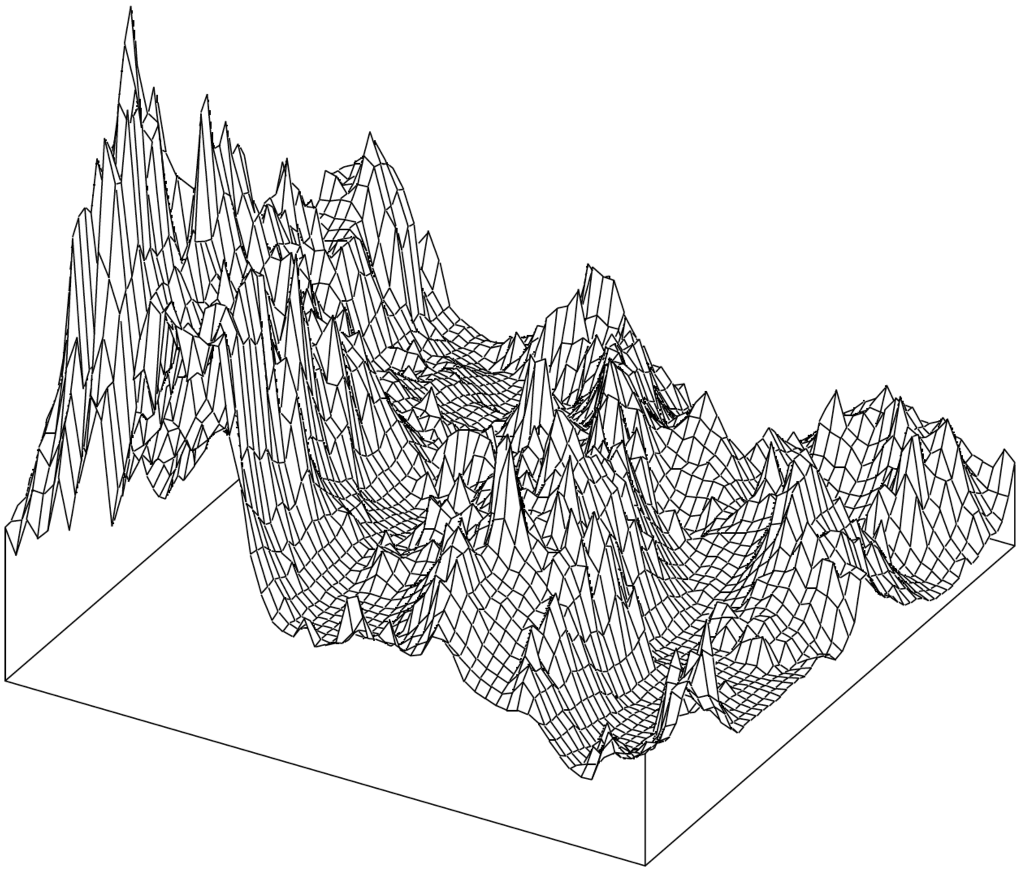


Application: Science

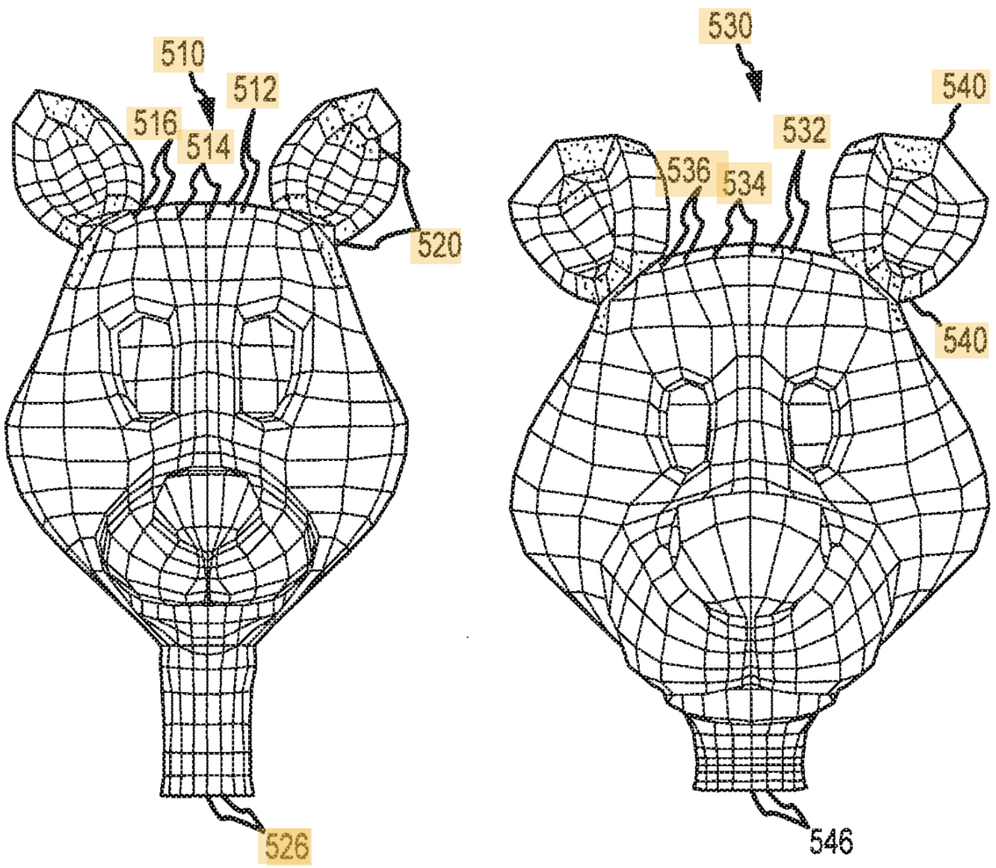


caffeine

Application: Geographic Information Systems (GIS)

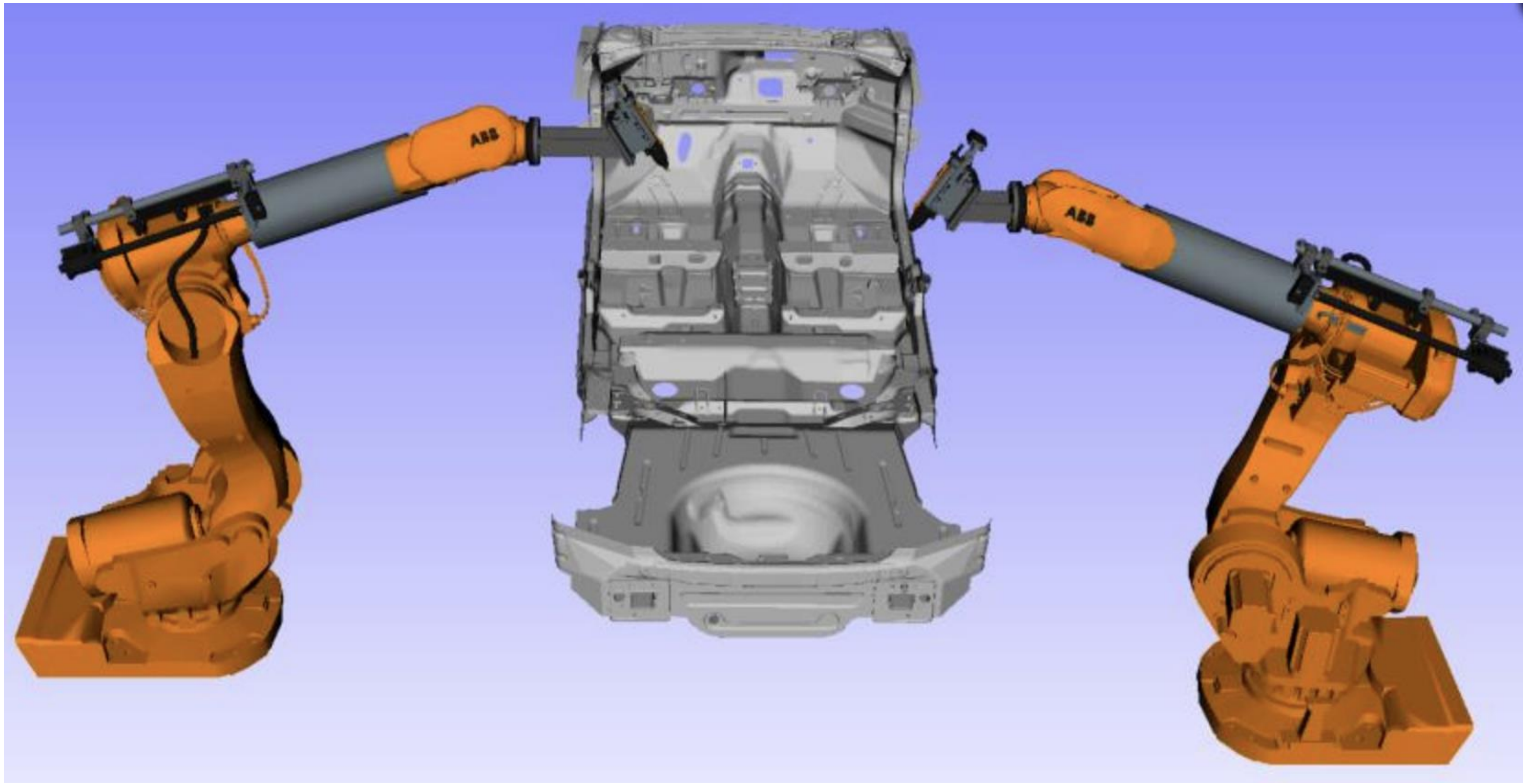


Application: Solid Modeling

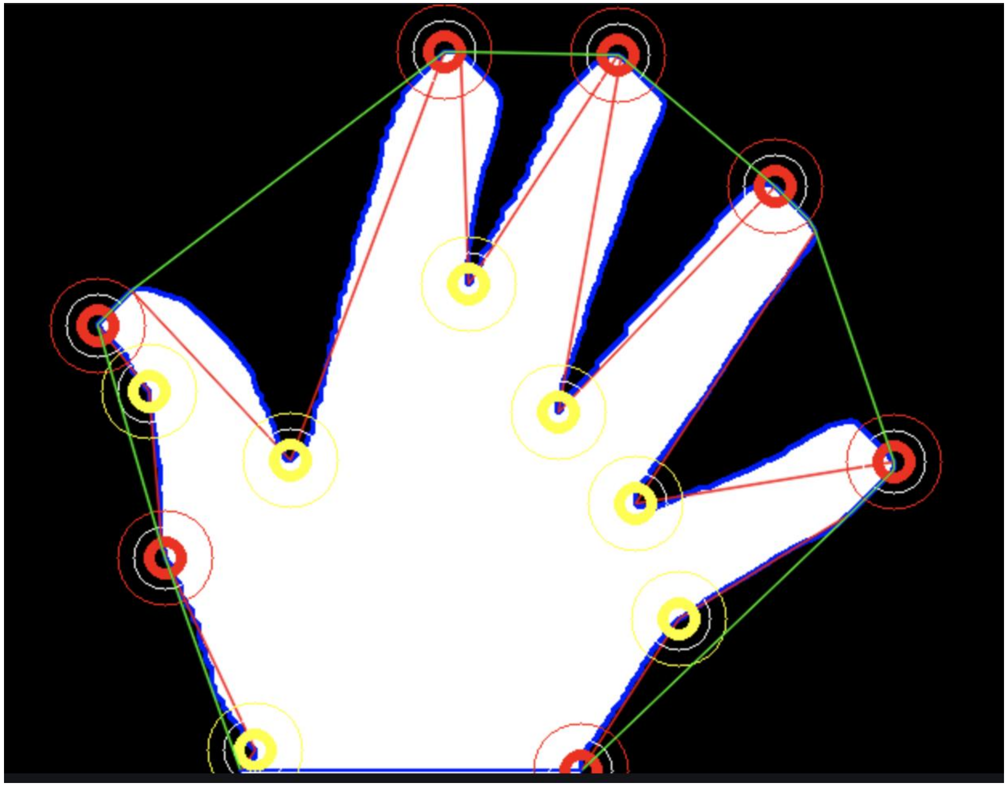


Application: Computer Graphics

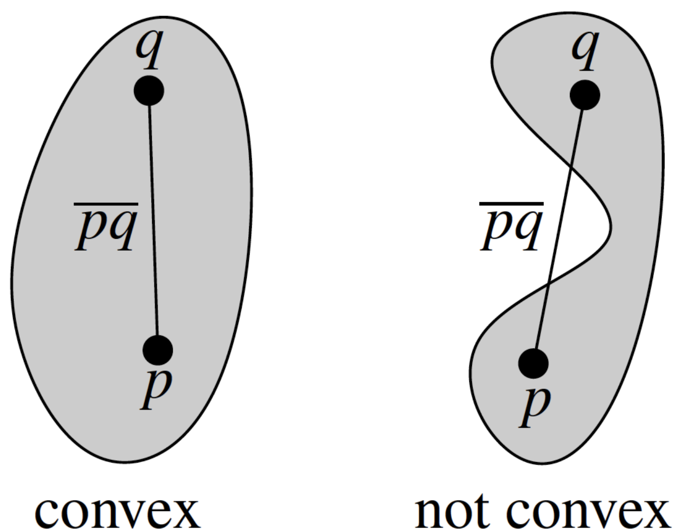
Application: Motion Planning and Robotics



Application: Shape Analysis and Computer Vision

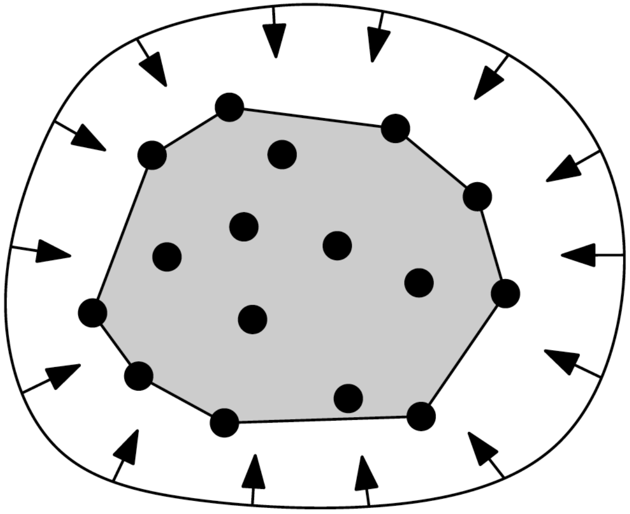


Convexity



Convex hull

- Smallest convex set containing all *n* points



Convex hull

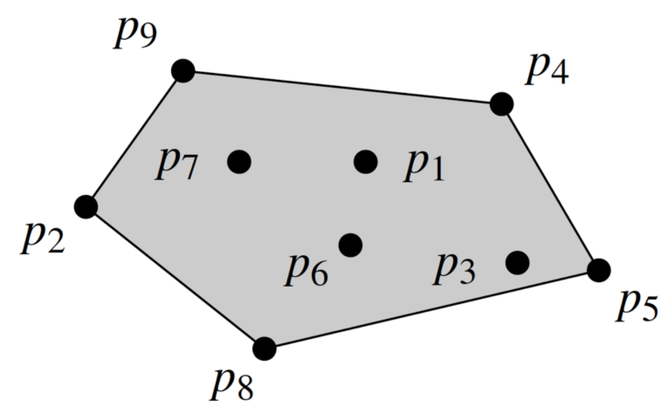
- Smallest convex set containing all *n* points

input = set of points:

$p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9$

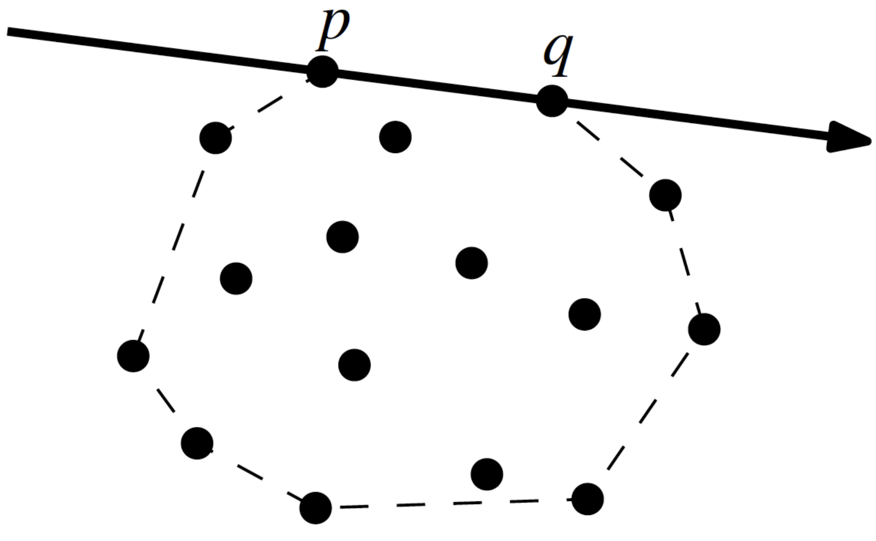
output = representation of the convex hull:

p_4, p_5, p_8, p_2, p_9



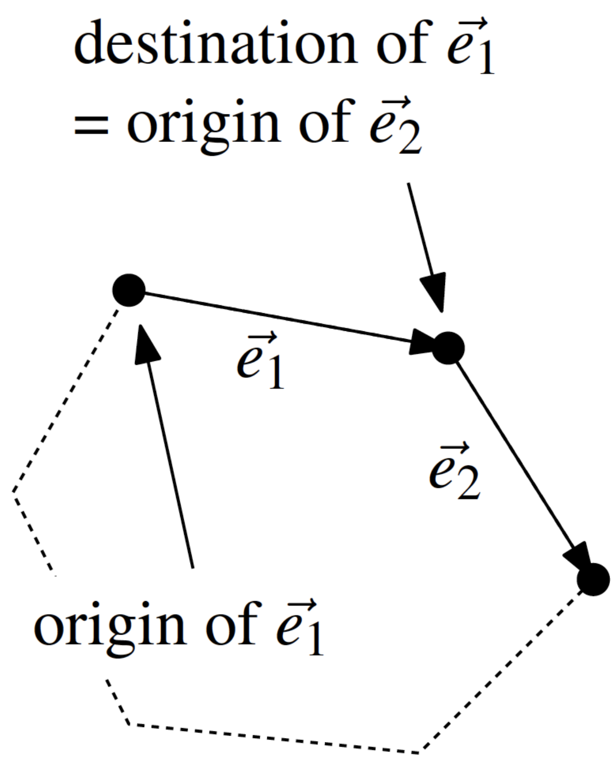
Useful Primitives

- Sidedness



Useful Primitives

- Orientation test: right turn or left turn



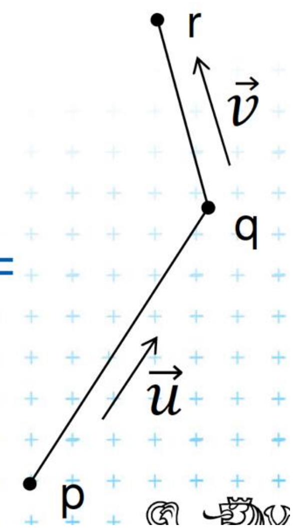
Computing Orientations

$$\text{orientation}(p,q,r) = \text{sign}\left(\det\begin{bmatrix}1 & p_x & p_y\\ 1 & q_x & q_y\\ 1 & r_x & r_y\end{bmatrix}\right) =$$

$$= \text{sign}\left((q_x - p_x)(r_y - p_y) - (q_y - p_y)(r_x - p_x)\right),$$

$$\text{where point } p = (p_x, p_y), \ldots$$

$$= \text{third coordinate of} = (\vec{u} \times \vec{v}),$$



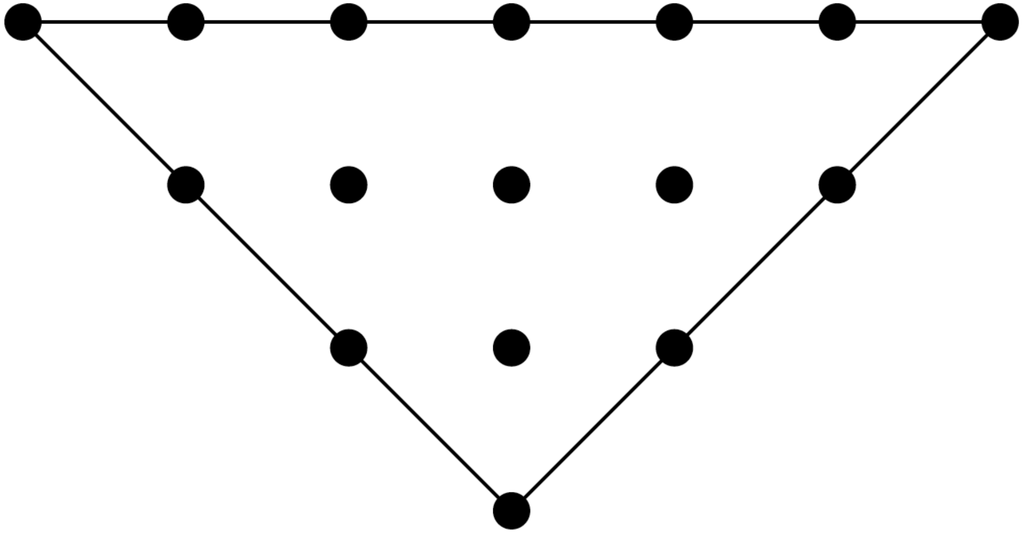
Three points	$\text{orientation}(p,q,r) =$
– lie on common line	= 0
– form a left turn	= +1 (positive)
– form a right turn	= –1 (negative)

A First Convex Hull Algorithm

Algorithm SlowConvexHull(P)
Input. A set P of points in the plane.
Output. A list $\mathcal{L}$ containing the vertices of $\mathcal{CH}(P)$ in clockwise order.

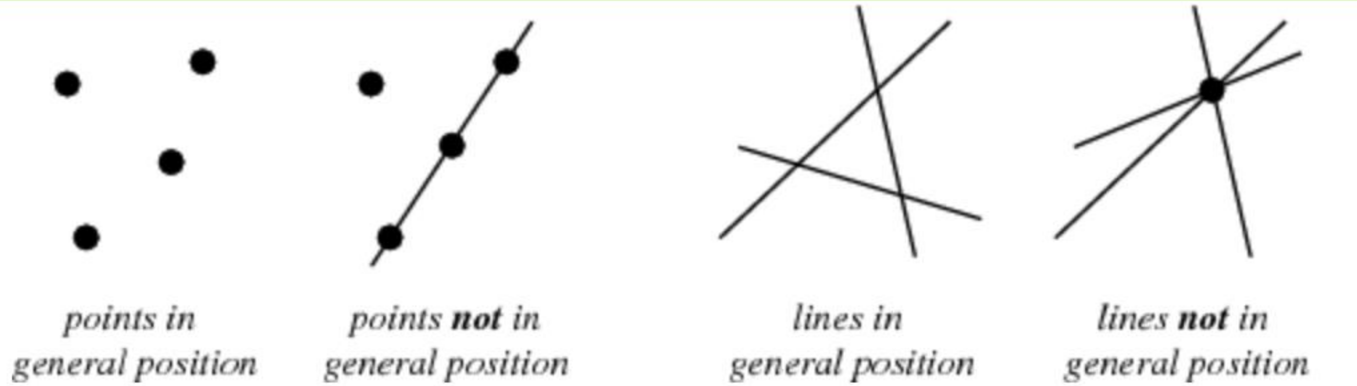
$E \leftarrow \emptyset$.
for all ordered pairs $(p,q) \in P \times P$ with p not equal to q
    do *valid* $\leftarrow$ **true**
        for all points $r \in P$ not equal to p or q
            do if r lies to the left of the directed line from p to q
                then *valid* $\leftarrow$ **false**.
        if *valid* **then** Add the directed edge $\overrightarrow{pq}$ to E.
From the set E of edges construct a list $\mathcal{L}$ of vertices of $\mathcal{CH}(P)$, sorted in clockwise order.

Degeneracies



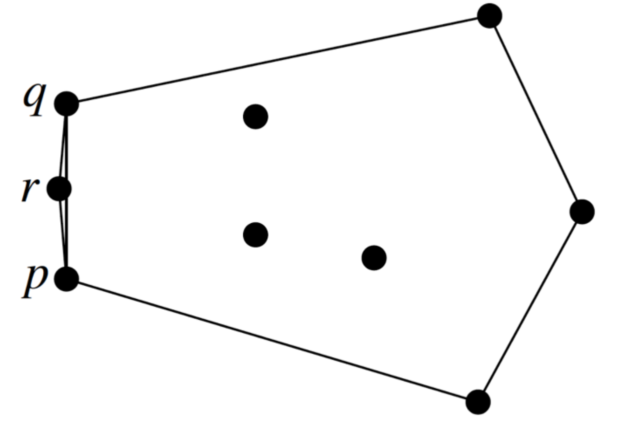
Dealing with Degeneracies

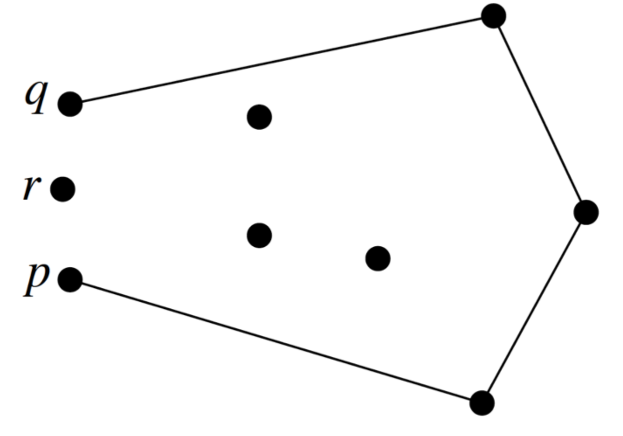
- Assume input is in **general position** and go back later to deal with degeneracies



Robustness

- Computing geometric primitives can introduce errors, e.g., if computations are done using floating point



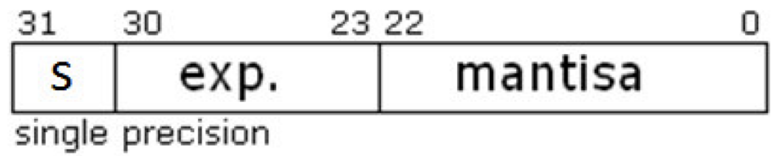


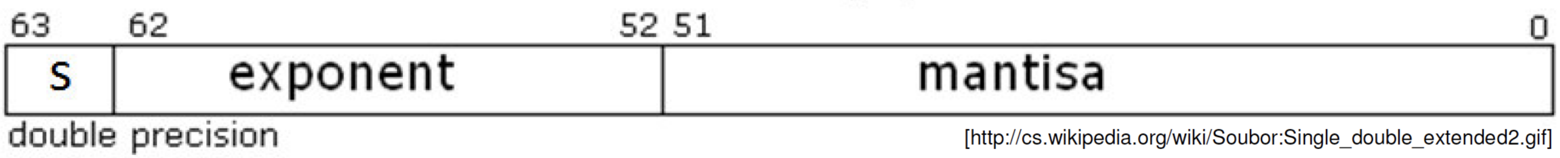
Floating Point Numbers are **not** exact

a) Limited numerical precision of real numbers

- Numbers represented as normalized

$$\pm m2^e$$



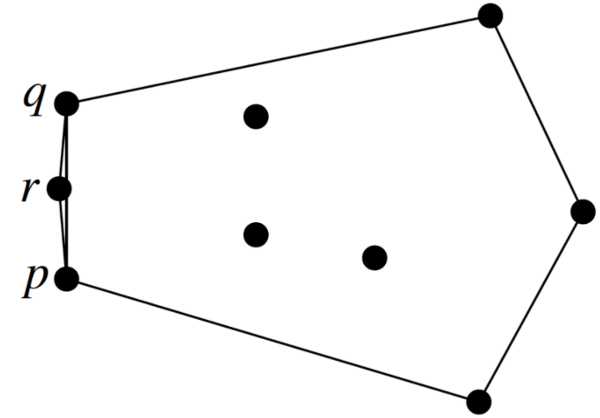


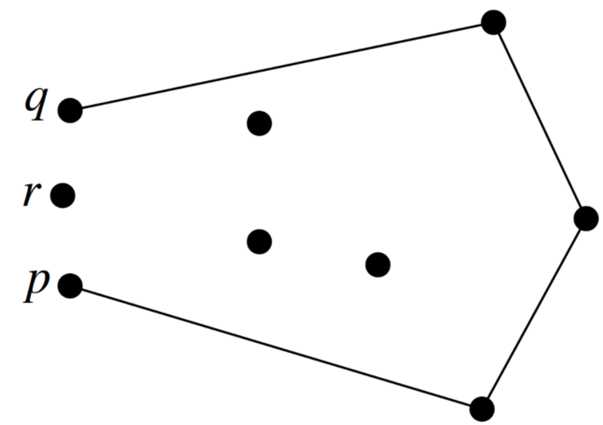
[http://cs.wikipedia.org/wiki/Soubor:Single_double_extended2.gif]

- The mantissa m is a 24-bit (53-bit) value whose most significant bit (MSB) is always 1 and is, therefore, not stored.
- Stored numbers (results) are <u>rounded</u> to 24/53 bits mantissa – lower bits are lost

Dealing with Robustness

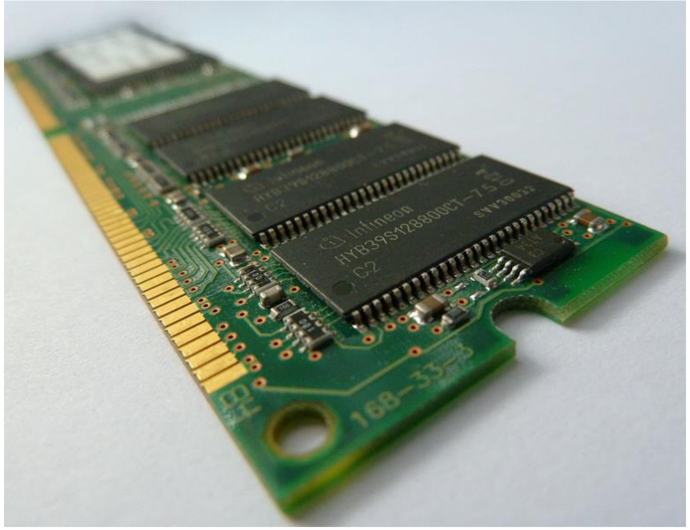
- Assume all arithmetic is **exact**.
  - Can be simulated using **floating point filters**, which we might discuss later if there is time
  - Gives rise to a computational model known as the **Real RAM**





Real RAM Model

- Not:



Real RAM Model

- A “Random Access Machine”:
  - a stored program
  - a computer memory: an array of cells
  - a central processing unit
  - Each memory cell or register can store a real number.
- Allowed operations include addition, subtraction, multiplication, and division, as well as comparisons.
- Some people also include things like square-roots and rounding to integers, but this is sometimes considered “cheating”.
- When analyzing algorithms for the real RAM, each allowed operation is typically assumed to take constant time.