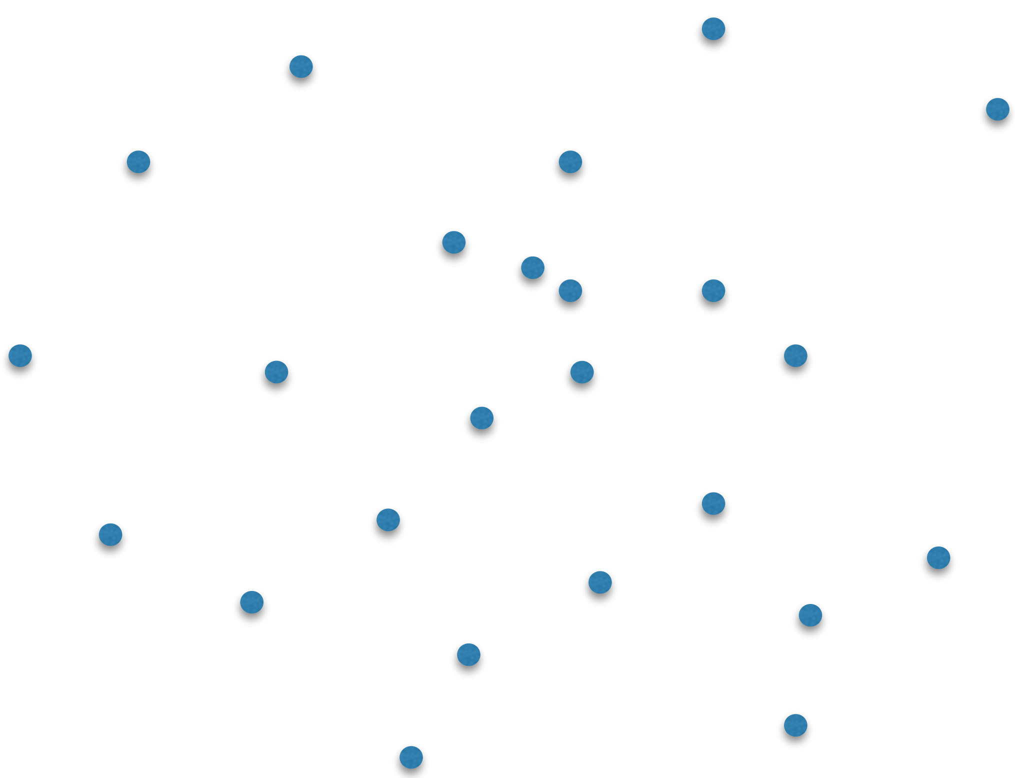
3D convex hulls

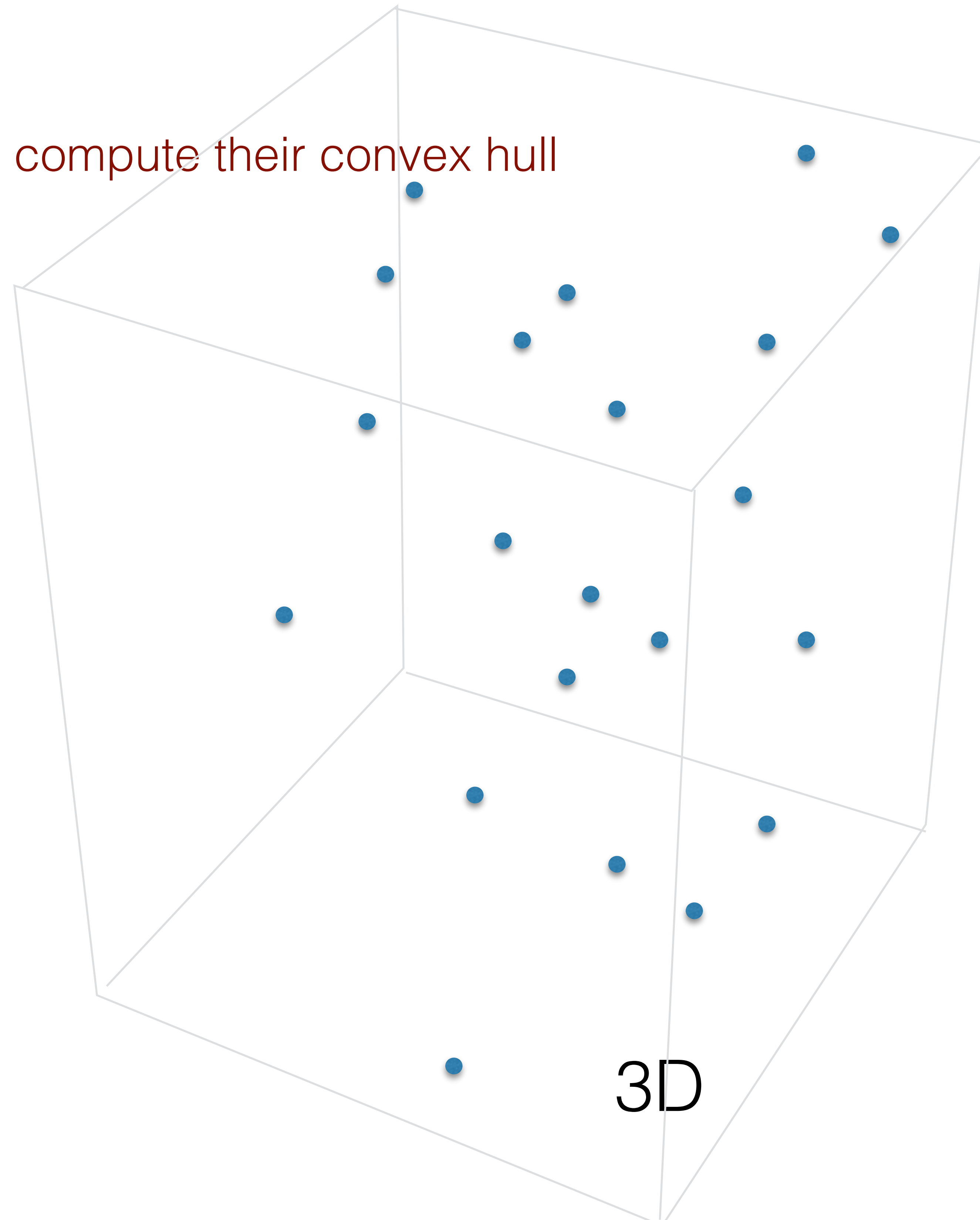
Computational Geometry [csci 3250]
Laura Toma
Bowdoin College

Convex Hulls

The problem: Given a set P of points, compute their convex hull

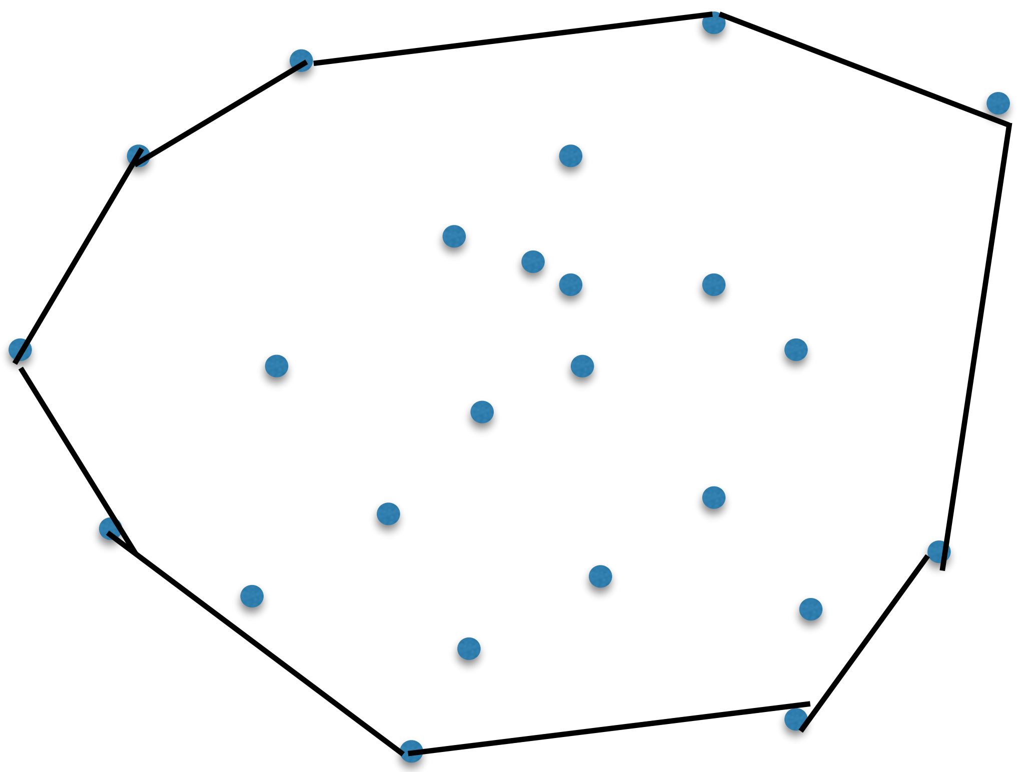


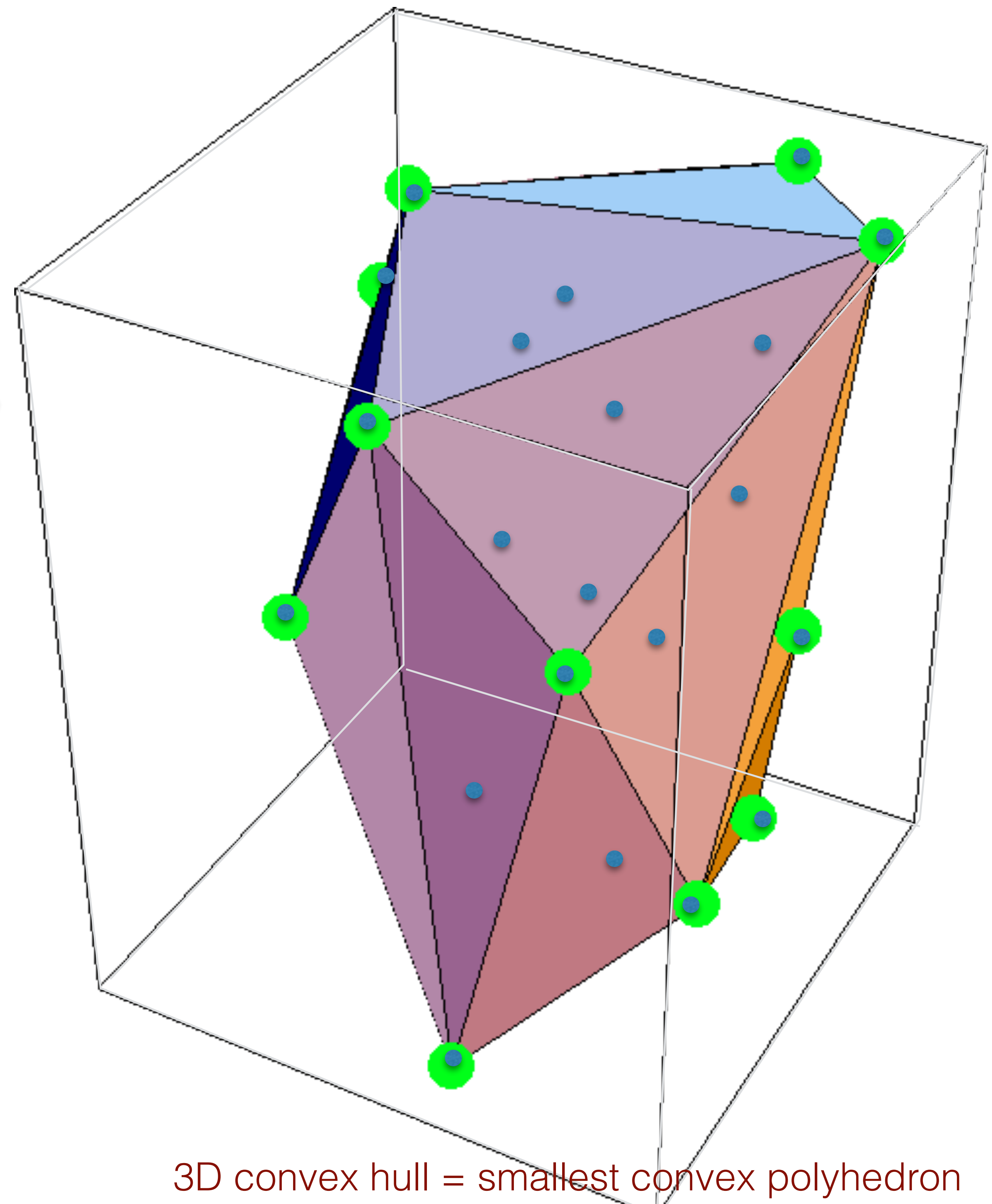
2D



3D

Convex Hulls

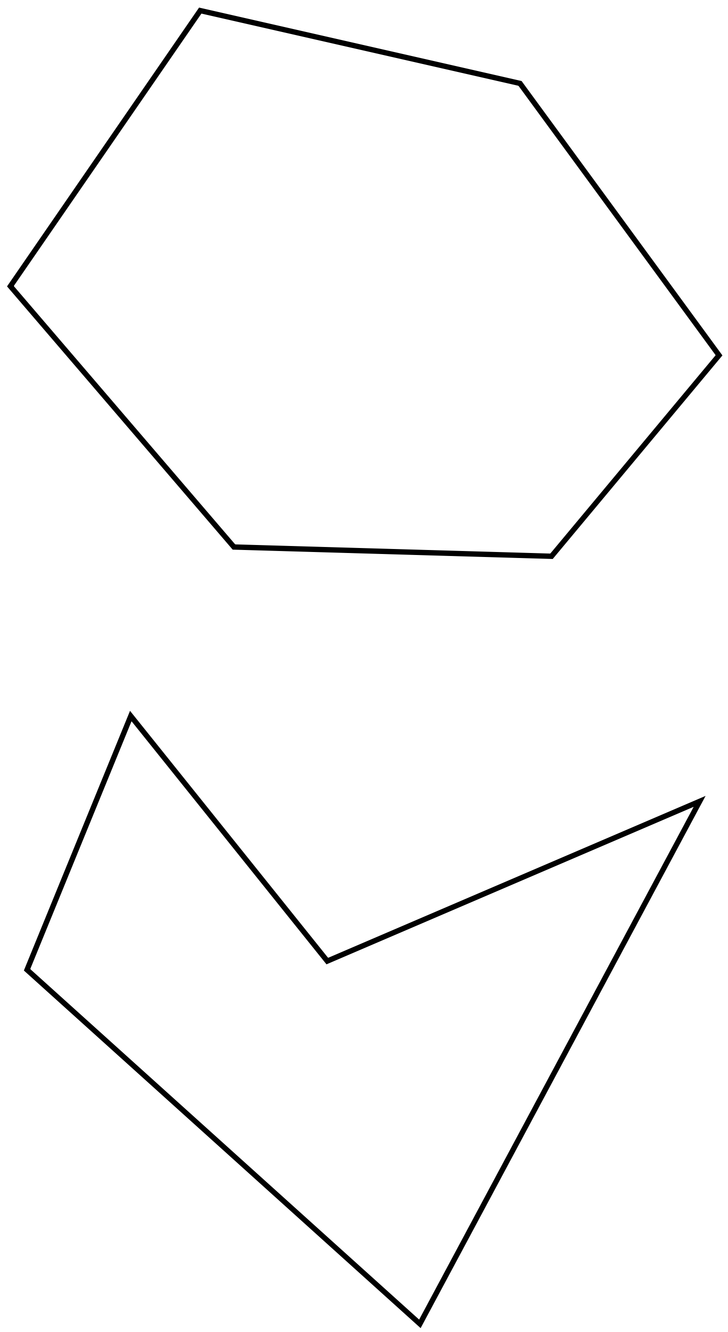




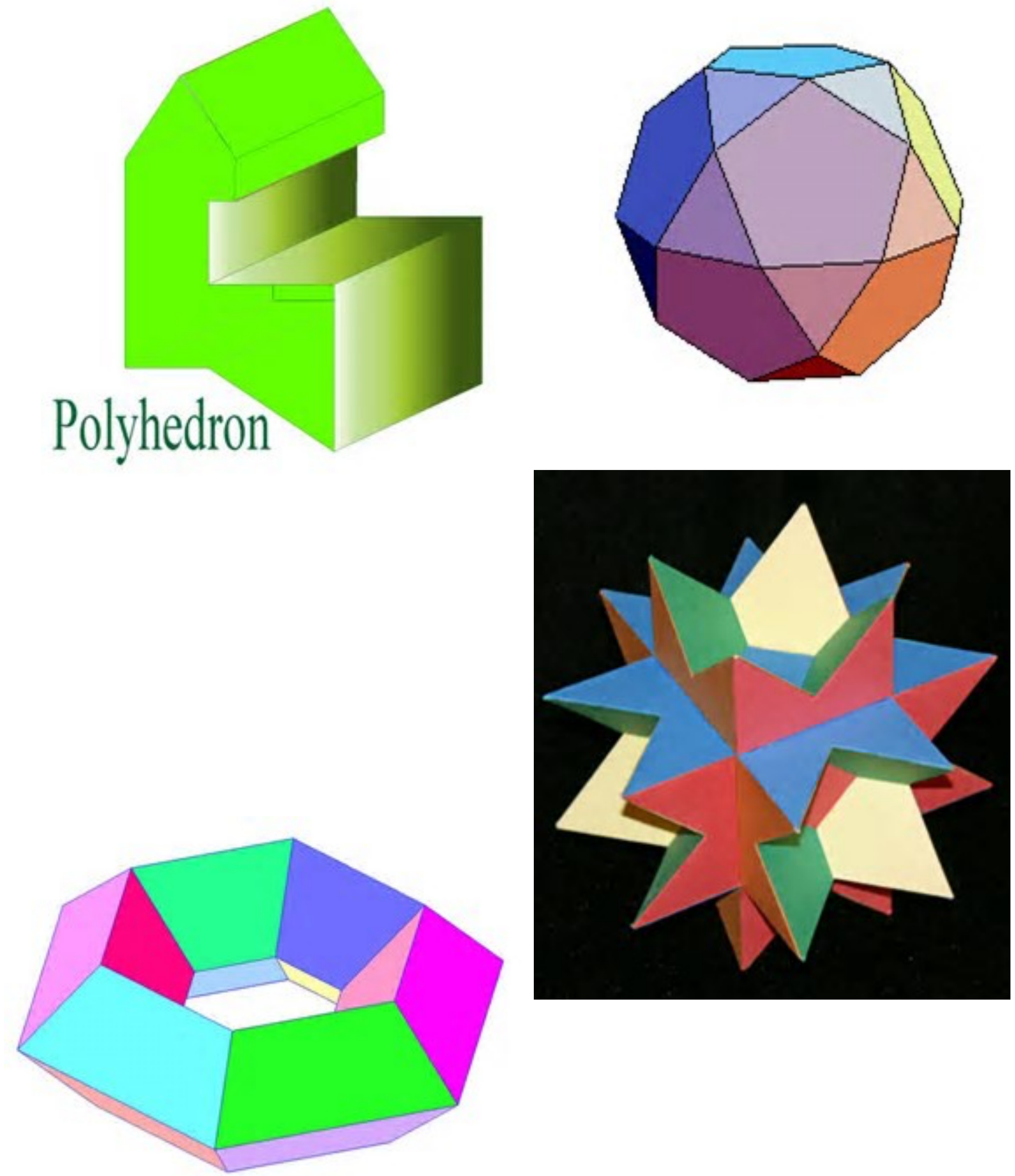
2D convex hull = smallest convex polygon (polytope) that contains P

3D convex hull = smallest convex polyhedron (polytope) that contains P

2D | **3D**



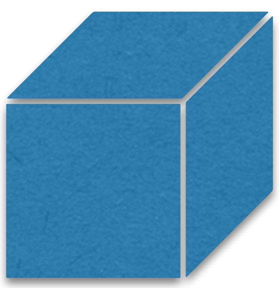
polygon

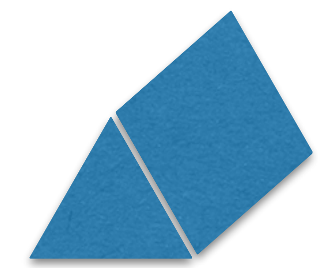


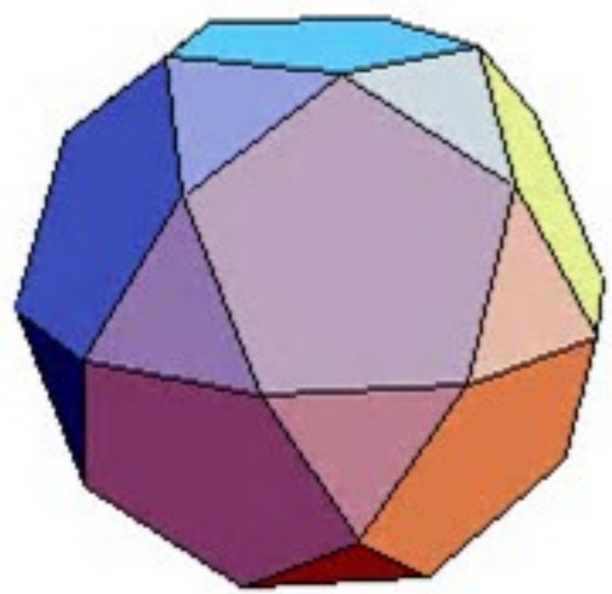
polyhedron

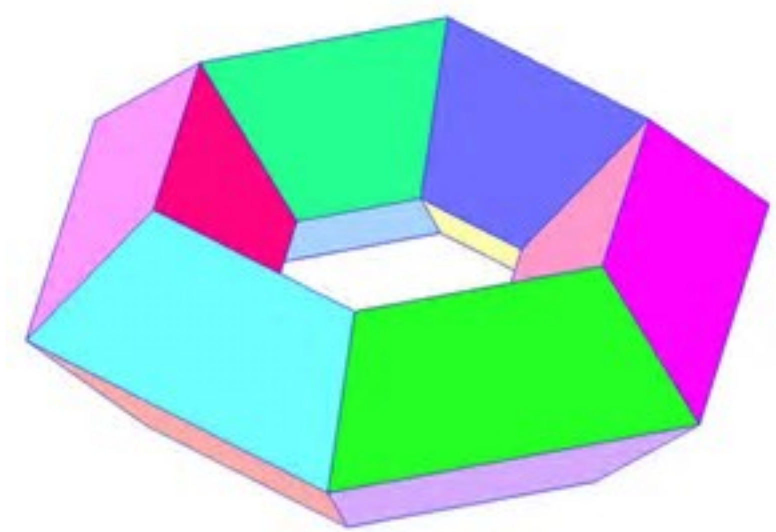
Polyhedron

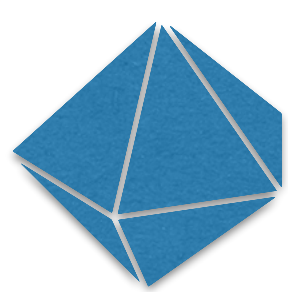
- region of space whose boundary consists of vertices, edges and (flat) faces, such that faces intersect properly
    - two faces are either disjoint; or
    - have a single vertex in common; or
    - have two vertices and the edge between them in common

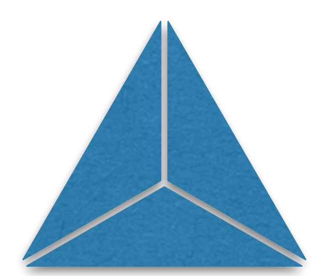












Polyhedra

- Also, local topology must be proper

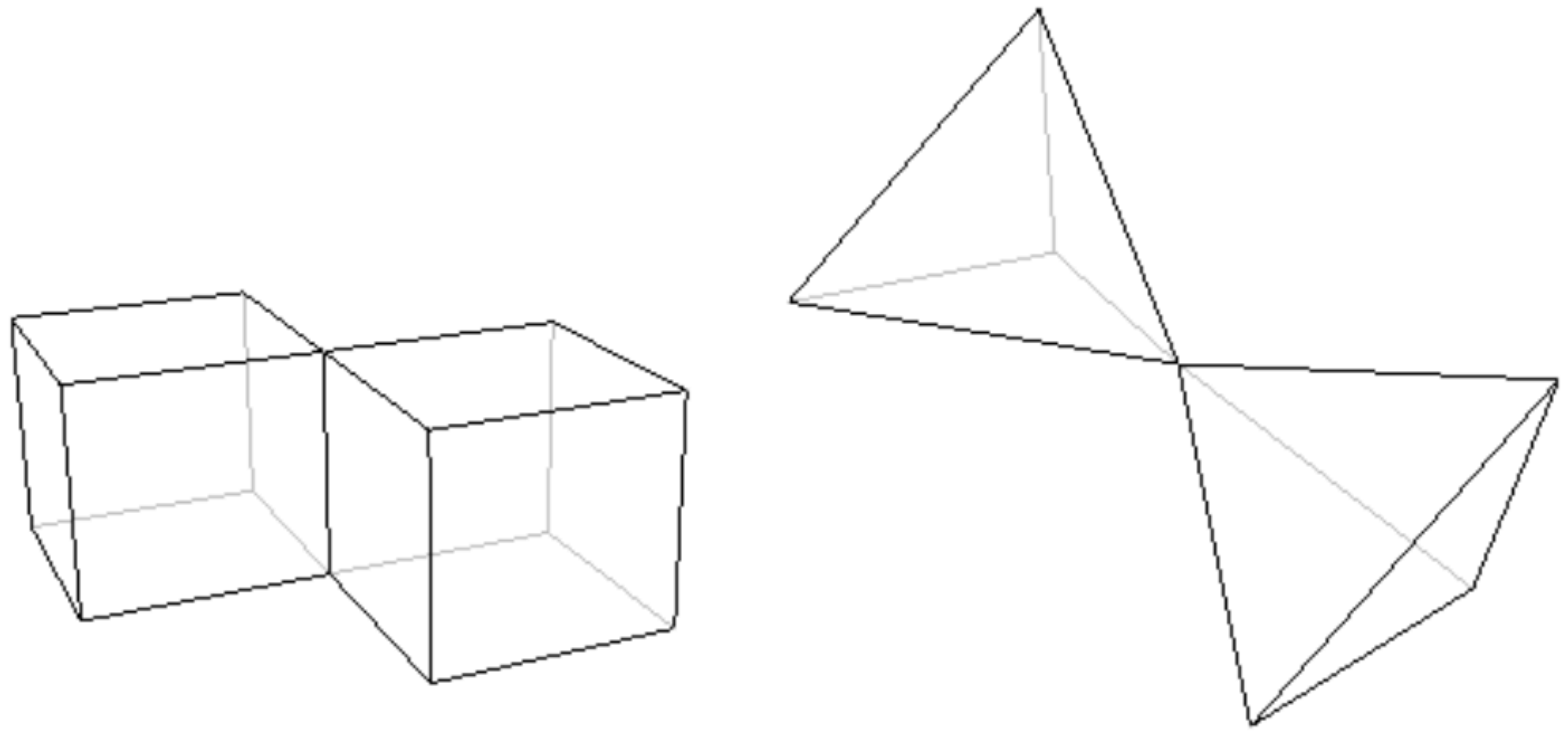
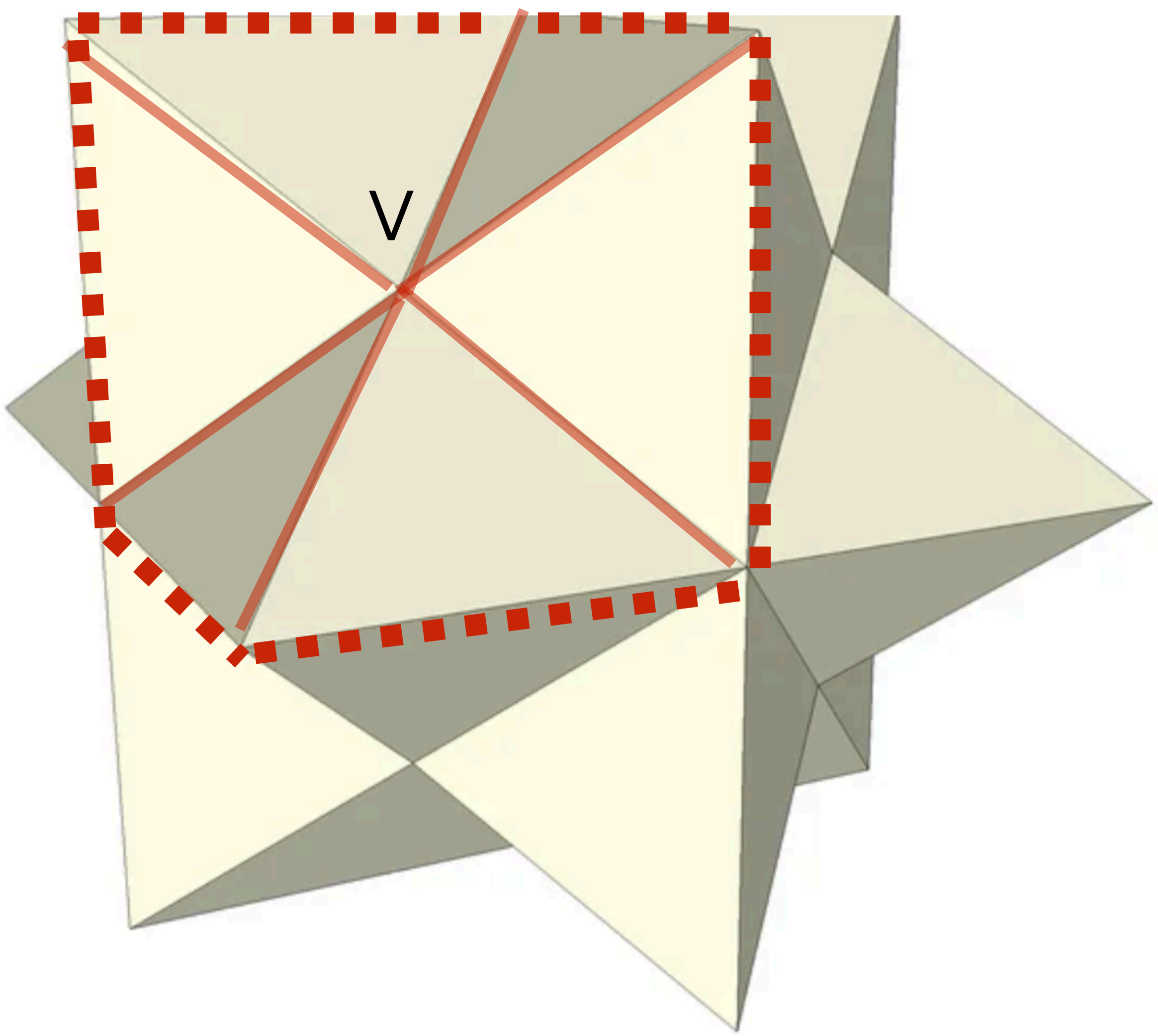
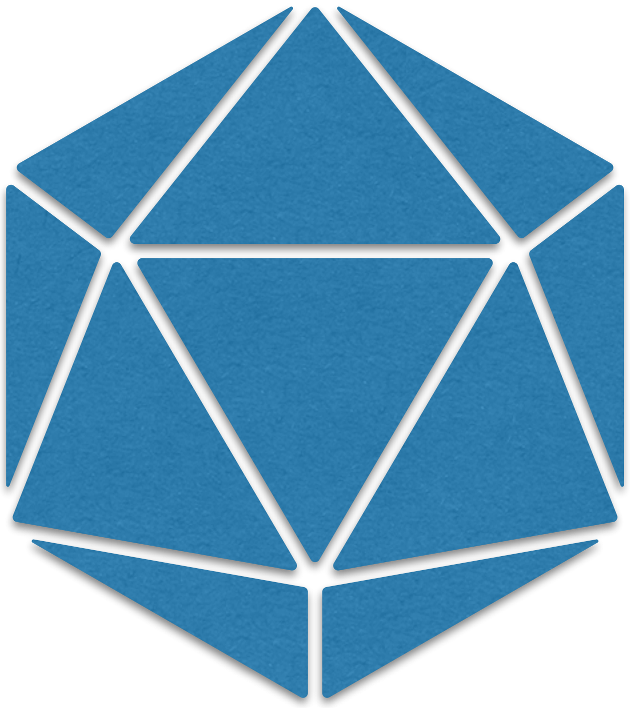


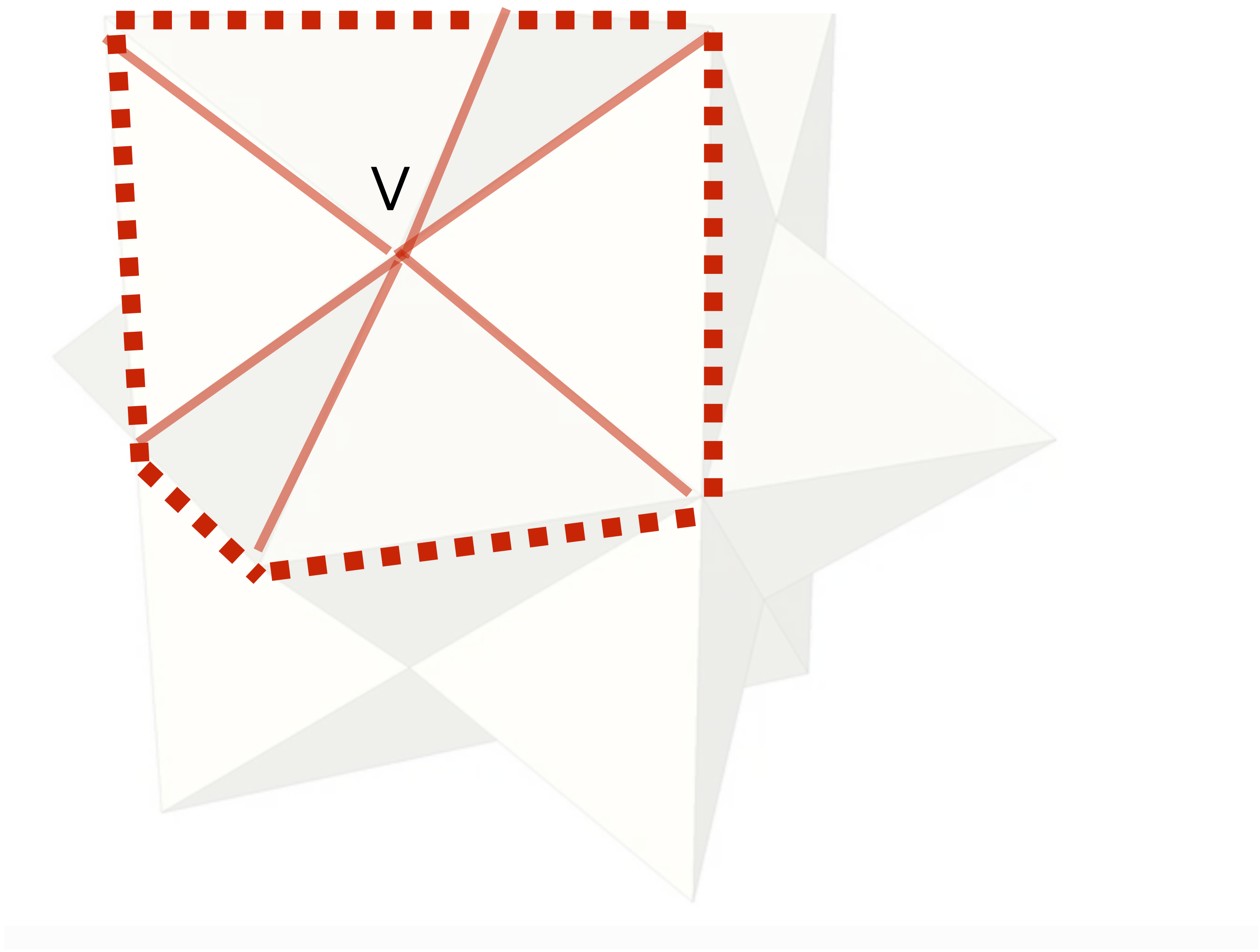
Figure 4: These objects are not polyhedra because they are made up of two separate parts meeting only in an edge (on the left) or a vertex (on the right).

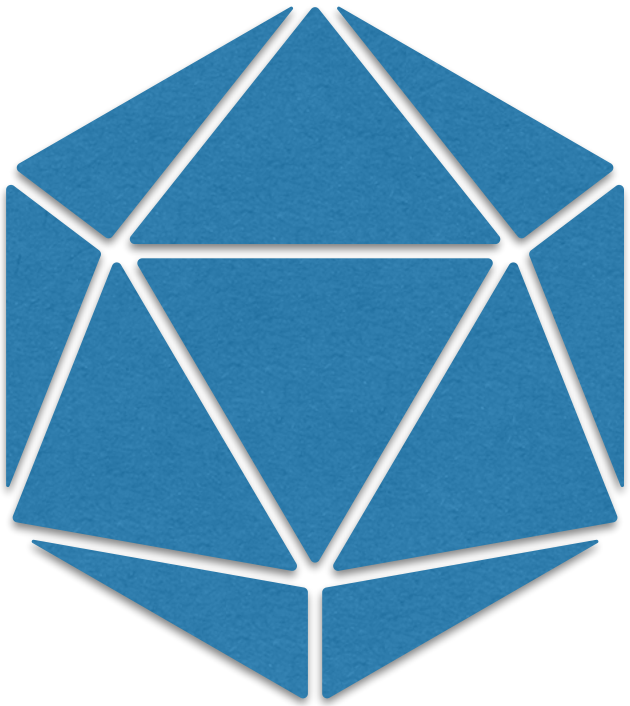
https://plus.maths.org/content/eulers-polyhedron-formula





The link of any vertex be is a simple, closet polygonal path.

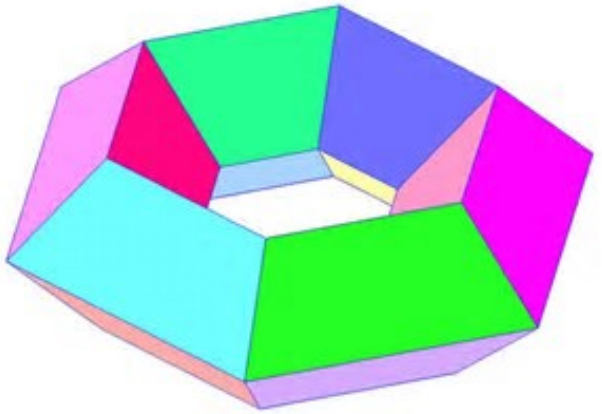




The link of any vertex be is a simple, closet polygonal path.

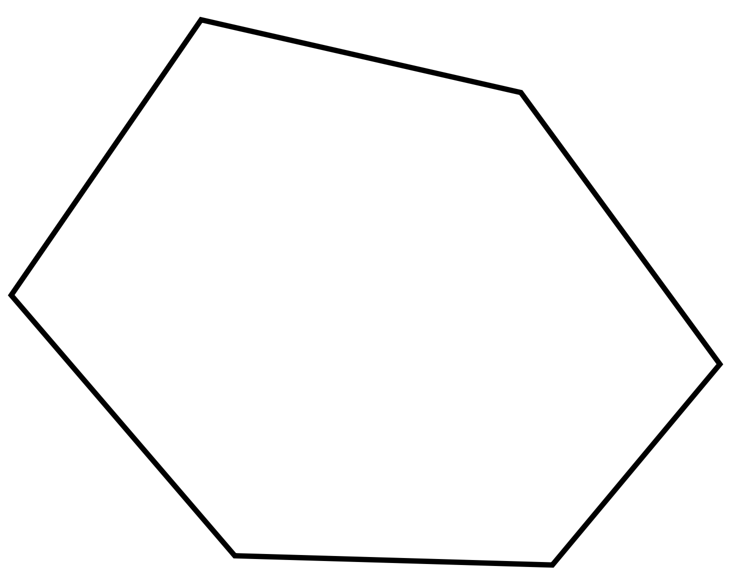
Polyhedra

- Also: global topology must be proper: surface is connected, closed and bounded.
  - Holes are allowed, as long as they don't disconnect
  - The nb of holes is called the genus of the surface

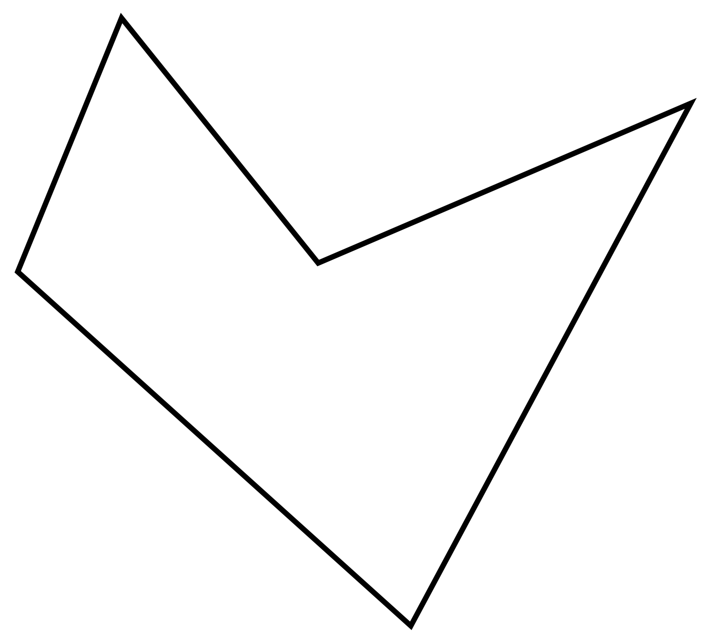


Convexity

A polygon P is **convex** if for any p, q in P, the segment pq lies entirely in P.



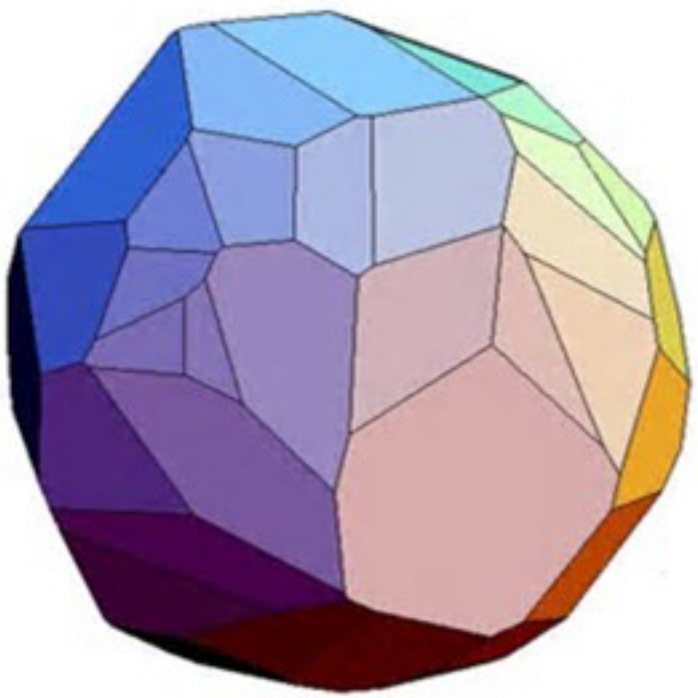
convex



non-convex

Convexity

A polyhedron P is **convex** if for any p, q in P, the segment pq lies entirely in P.



convex

non-convex

convex polyhedron : polytop

digression start

Regular polygons in 2D

- A regular polygon has equal sides and angles

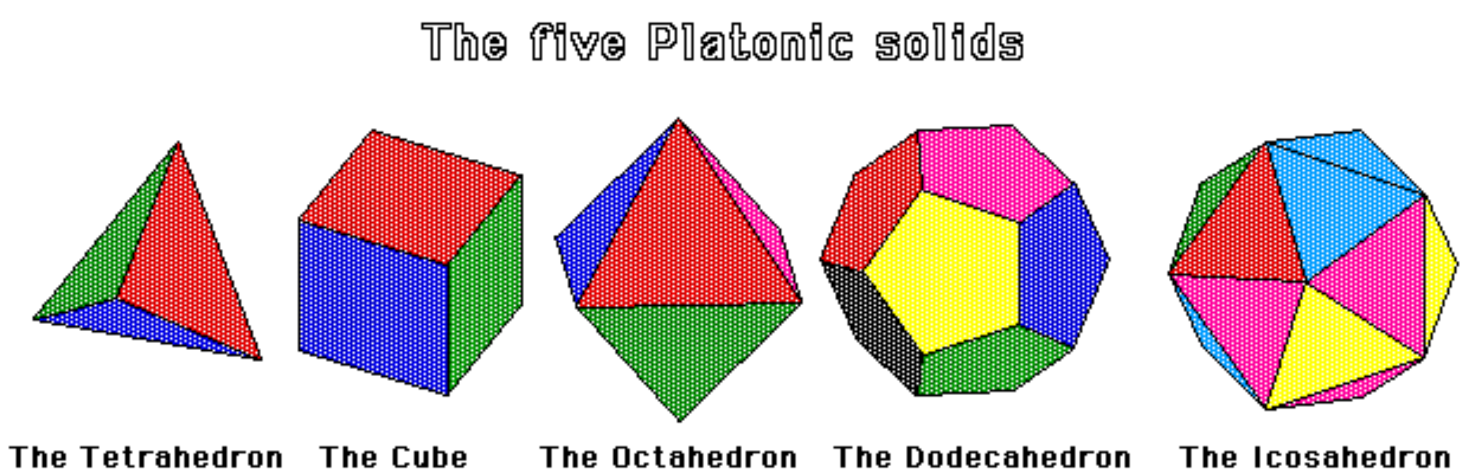
Set of convex regular n-gons

Regular polygons

Regular polytops in 3D

- Regular polytop:
  - faces are congruent regular polygons
  - the number of faces incident to each vertex is the same (and equal angles)

Surprisingly, there exist only 5 regular polytops



The five regular solids discovered by the Ancient Greek mathematicians are:

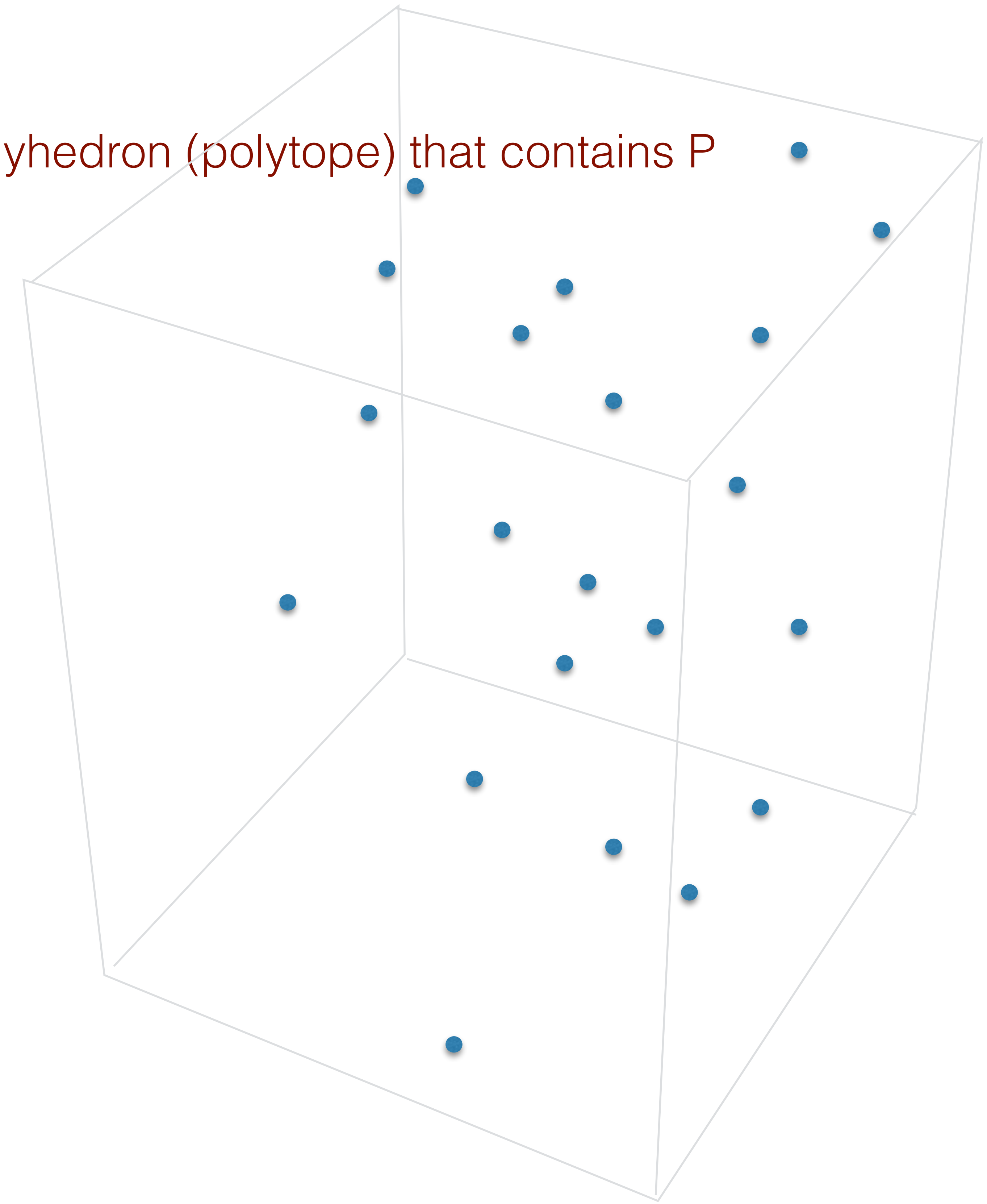
The **Tetrahedron**:	4 vertices	6 edges	4 faces	each with 3 sides
The **Cube**:	8 vertices	12 edges	6 faces	each with 4 sides
The **Octahedron**:	6 vertices	12 edges	8 faces	each with 3 sides
The **Dodecahedron**:	20 vertices	30 edges	12 faces	each with 5 sides
The **Icosahedron**:	12 vertices	30 edges	20 faces	each with 3 sides

The solids are regular because the same number of sides meet at the same angles at each vertex and identical polygons meet at the same angles at each edge.
These five are the only possible regular polyhedra.

digression end

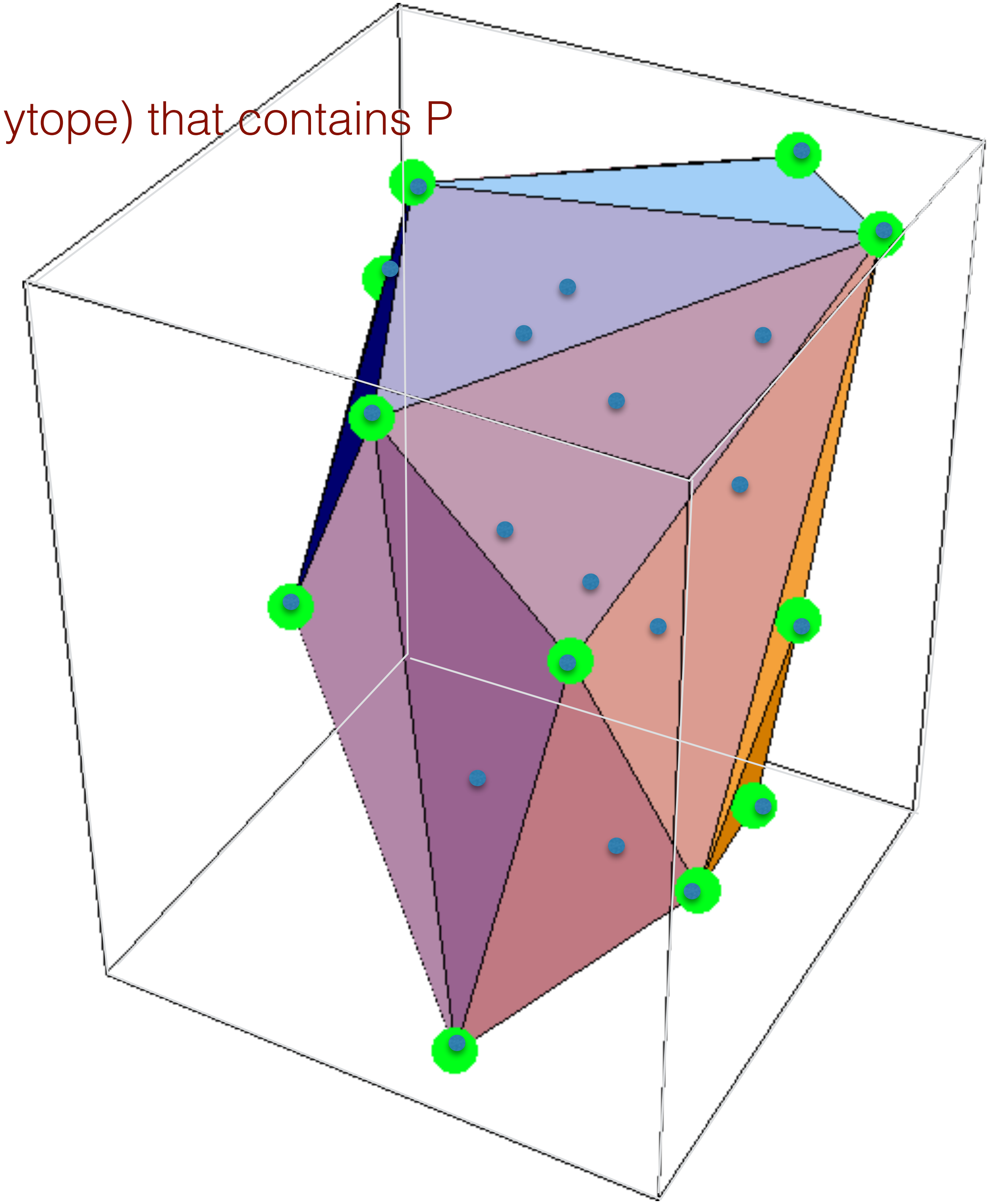
Convex Hulls in 3D

3D convex hull = smallest convex polyhedron (polytope) that contains P



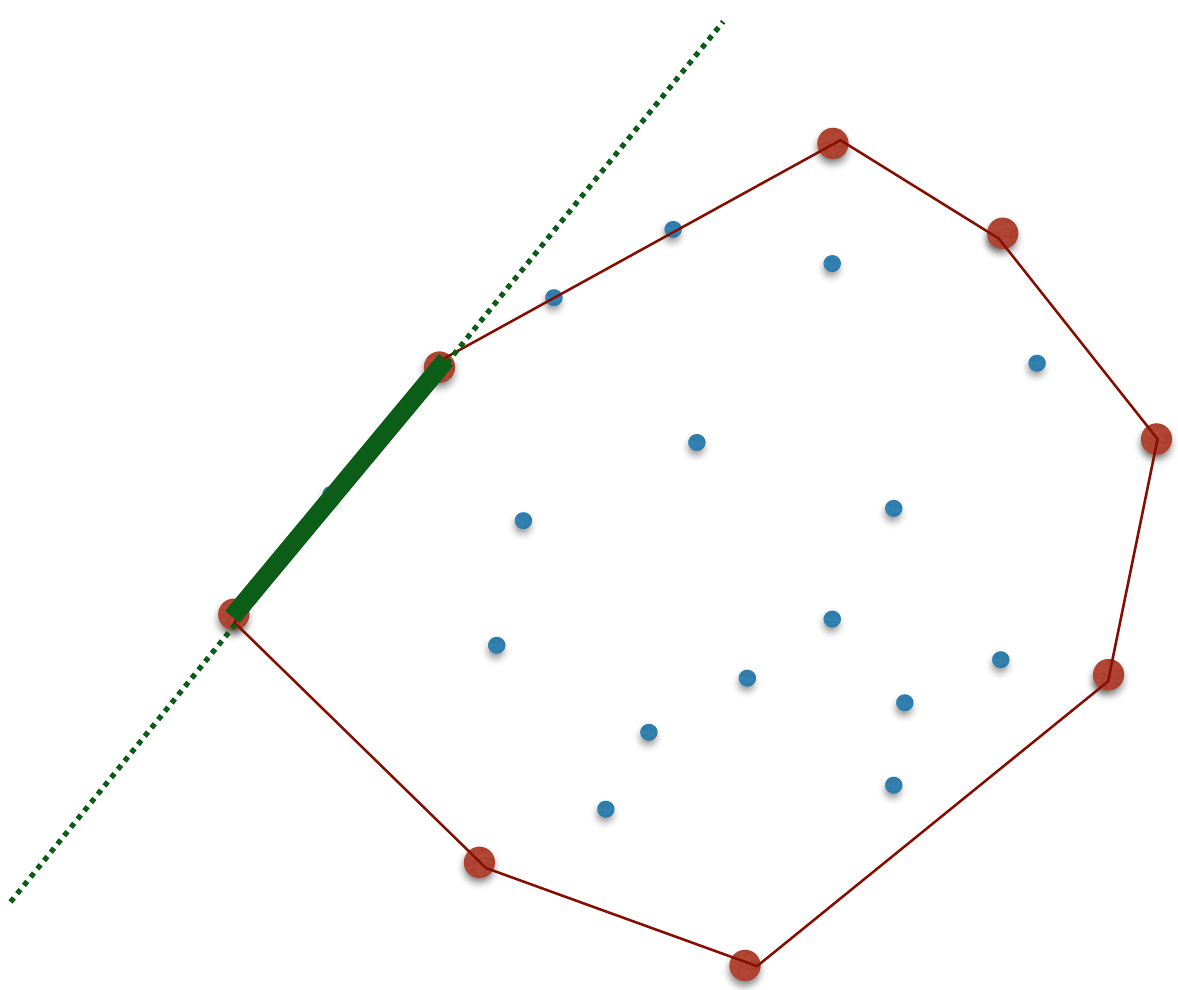
Convex Hulls in 3D

The smallest convex polyhedron (polytope) that contains P



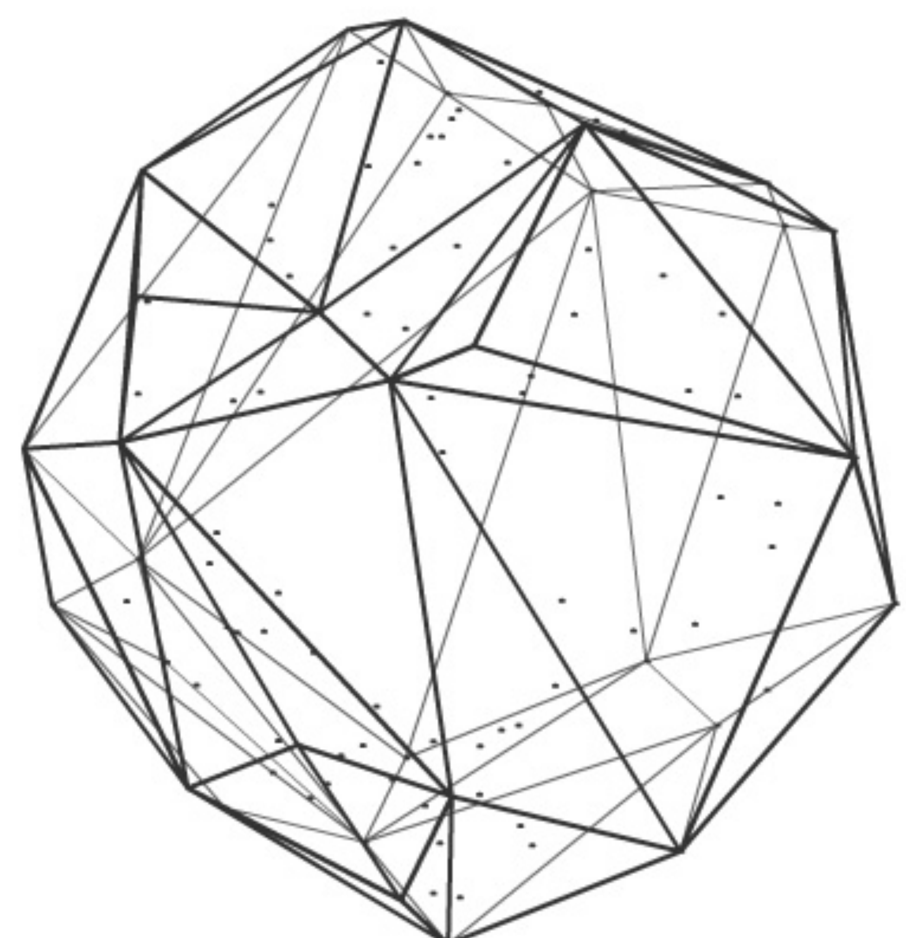
Properties of **2d** hull

- 2d hull consists of all extreme edges and vertices
- All internal angles are < 180
- Walking counterclockwise—> left turns
- Points on hull are sorted in radial order wrt a point inside

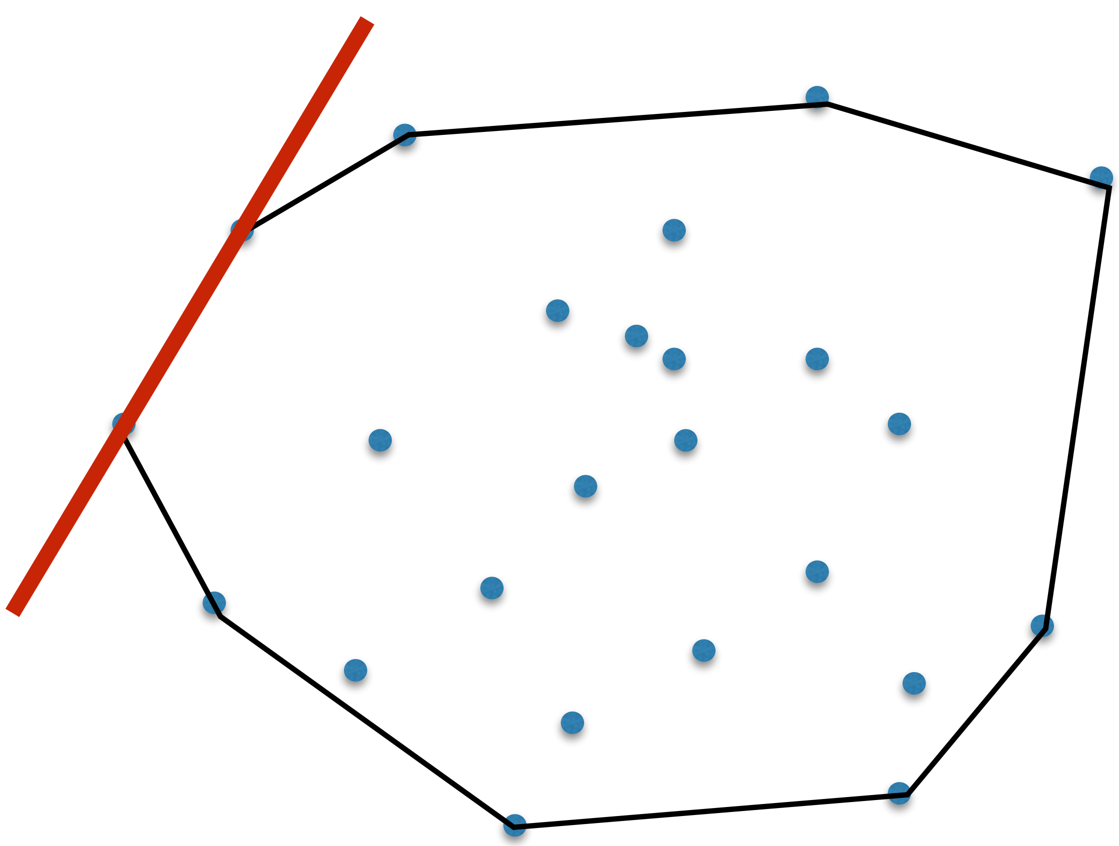


Properties of **3d** hull

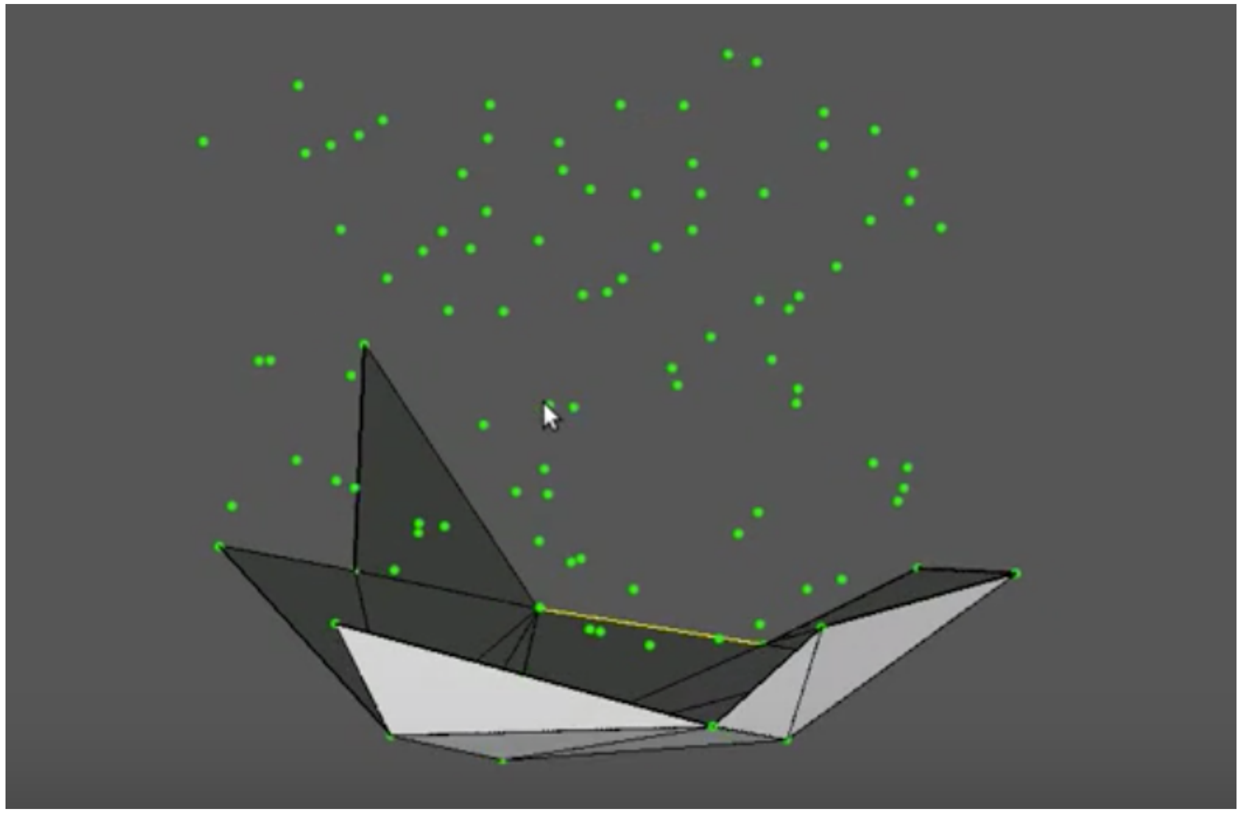
- 3d hull consists of all faces, edges and vertices
- All internal angles between faces are < 180
- ~~Walking counterclockwise—> left turns~~
- ~~Points on CH are sorted in radial order wrt a point inside~~



Faces, edges, vertices on the hull are **extreme**.



2D



3D

Computing the Hull

2D		3D
Naive	$O(n^3)$	
Gift wrapping	O(nh)	
Graham scan	O(n lg n)	does not extend..
Quickhull	O(n lg n), $O(n^2)$	
Incremental	O(n lg n)	
Divide-and-conquer	O(n lg n)	

Lower bound in 3D: $\Omega(n \lg n)$ ⟵ Is this achievable?

Naive 3d hull

3d hull: Naive algorithm

Algorithm idea:

- For every triplet of points (pi,pj,pk):
  - check if plane defined by it is extreme
  - if it is, add it to the list of CH faces

- Sketch how to determine if a triplet is extreme and analyze it

  is_extreme(point3d a, point3d b, point3d c, vector<point3d> P)

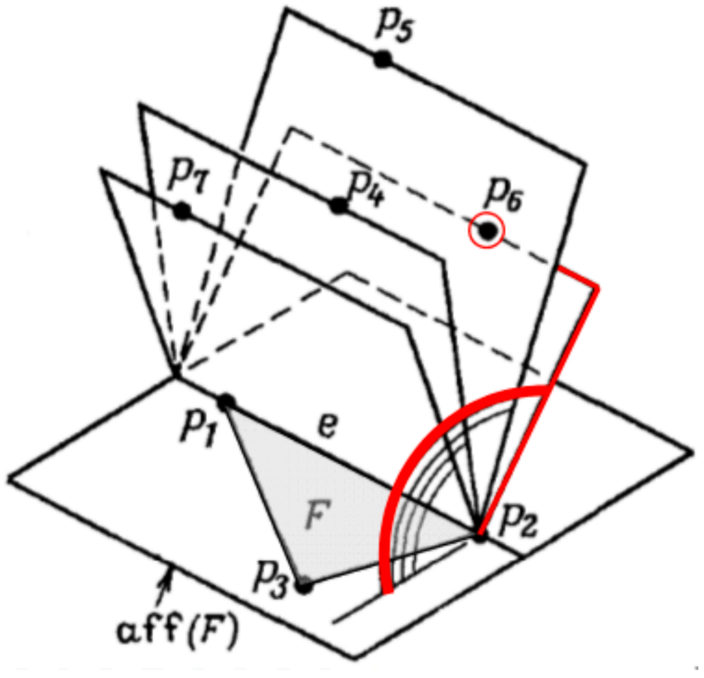
Gift wrapping

3d hull: Gift wrapping

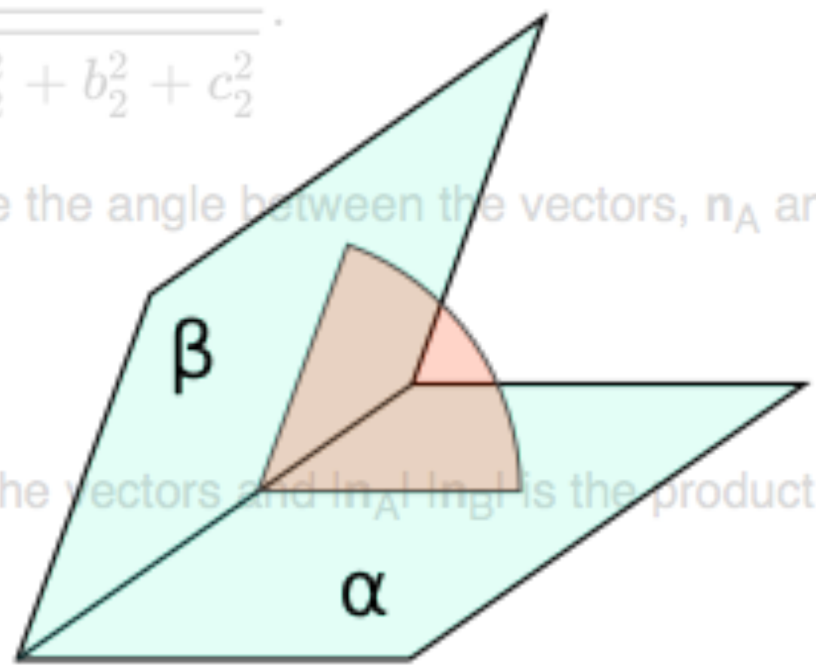
Algorithm

- find a face guaranteed to be on the CH
- REPEAT
  - find an edge e of a face f that's on the CH, and such that the face on the other side of e has not been found.
  - for all remaining points pi, find the angle of (e,pi) with f
  - find point pi with the minimal angle; add face (e,pi) to CH

- Analysis: O(n x F), where F is the number of faces on CH



A **dihedral angle** is the angle between two intersecting planes.



Angle between two planes (α, β, green) in a third plane (pink) which cuts the line of intersection at right angles

Mathematical background [edit]

When the two intersecting planes are described in terms of Cartesian coordinates by the two equations

$$a_1 x + b_1 y + c_1 z + d_1 = 0$$
$$a_2 x + b_2 y + c_2 z + d_2 = 0$$

the dihedral angle, φ between them is given by:

$$\cos\varphi = \frac{|a_1 a_2 + b_1 b_2 + c_1 c_2|}{\sqrt{a_1^2 + b_1^2 + c_1^2}\sqrt{a_2^2 + b_2^2 + c_2^2}}.$$

An alternative method is to calculate the angle between the vectors, $\mathbf{n}_A$ and $\mathbf{n}_B$, which are normal to the planes.

$$\cos\varphi = \frac{|\mathbf{n}_A \cdot \mathbf{n}_B|}{|\mathbf{n}_A||\mathbf{n}_B|}$$

where $\mathbf{n}_A \cdot \mathbf{n}_B$ is the dot product of the vectors and $|\mathbf{n}_A| |\mathbf{n}_B|$ is the product of their lengths. [1]

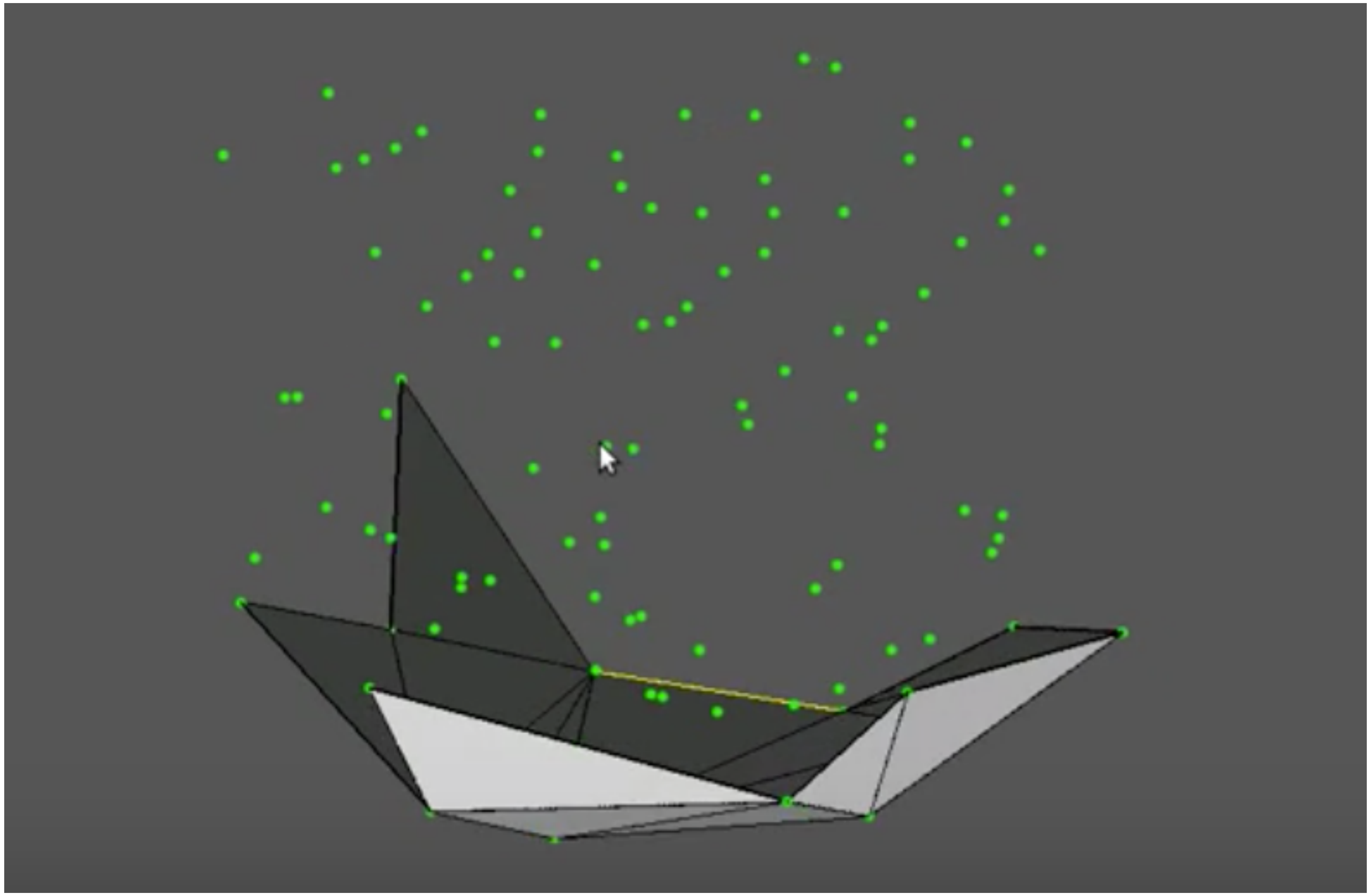
3d hull: Gift wrapping

Algorithm

- find a face guaranteed to be on the CH
- REPEAT
  - find an edge e of a face f that's on the CH, and such that the face on the other side of e has not been found.
  - for all remaining points pi, find the angle of (e,pi) with f
  - find point pi with the minimal angle; add face (e,pi) to CH

- To think
  - finding first face?
  - How to keep track of the hull? we'll need to store the connectivity (what faces are adjacent, for an edge which faces its adjacent to, etc)
  - How to keep track of the boundary of the hull (the edges that have only one face discovered)?

Gift wrapping in 3D



- YouTube
  - Video of CH in 3D (by Lucas Benevides)

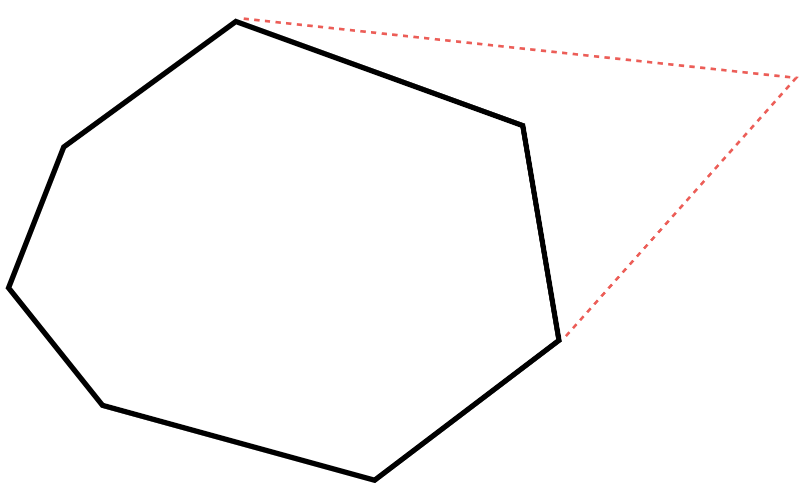
From 2D to 3D

2D		3D
Naive	$O(n^3)$	$O(n^4)$
Gift wrapping	O(nh)	$O(n \times F)$
Graham scan	O(n lg n)	does not extend to 3D
Quickhull	O(n lg n), $O(n^2)$	
Incremental	O(n lg n)	
Divide-and-conquer	O(n lg n)	

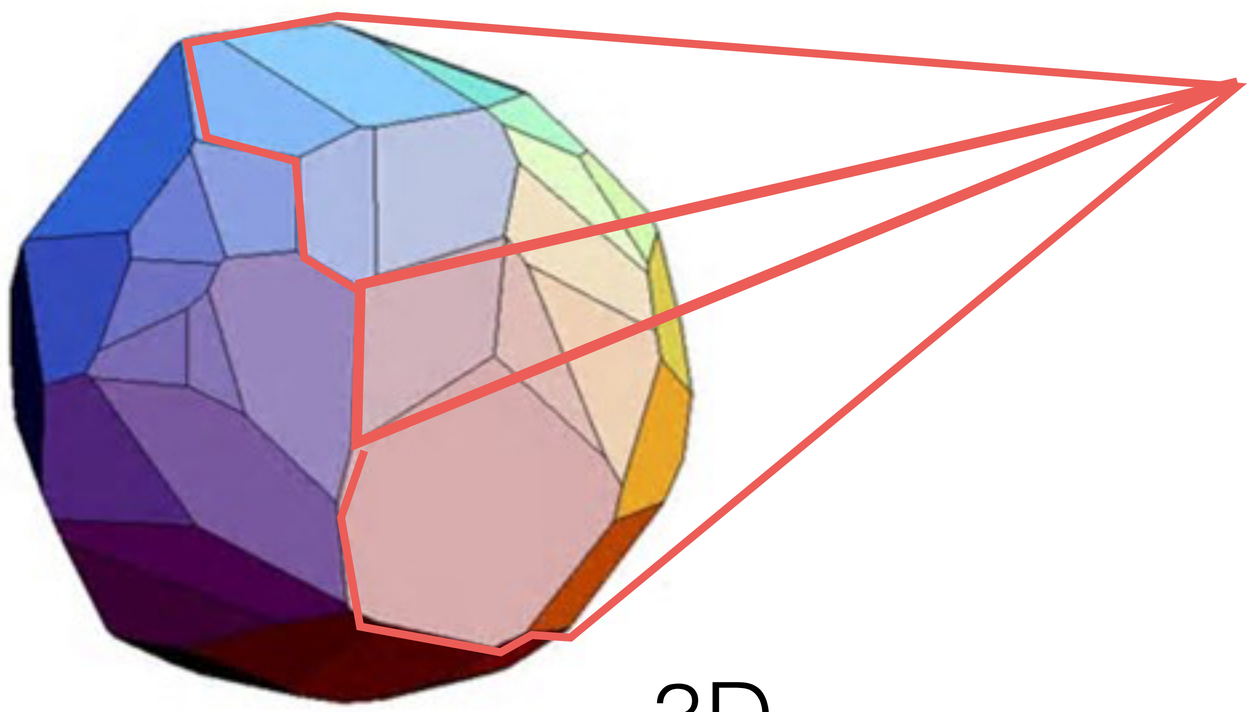
Incremental 3D hull

Incremental 3d hull

- sort points lexicographically
- initialize hull H = {p1,p2,p3}
- for i= 4 to n
    - //invariant: H represents the CH of $p_1..p_{i-1}$
    - add p_i to H and update H to represent the CH of $p_1..p_i$

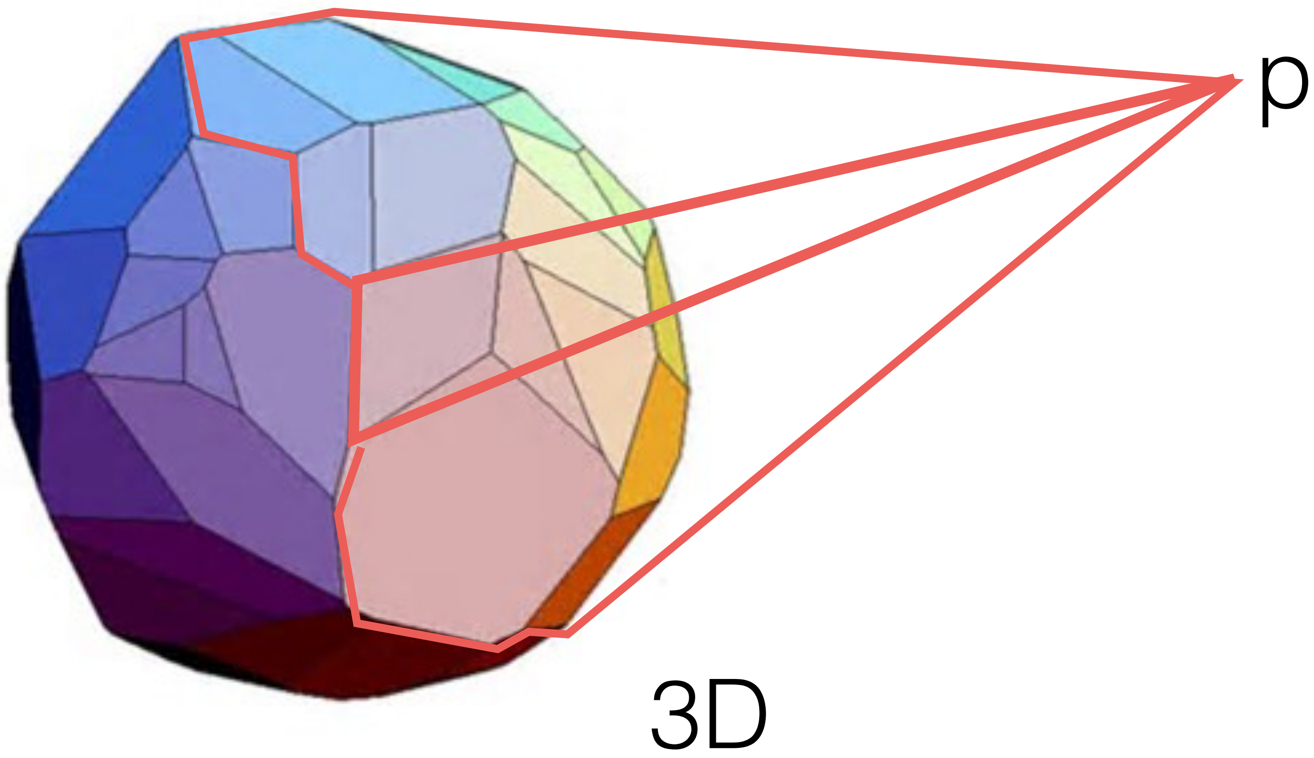


2D



3D

Incremental 3d hull



Imagine standing at p and looking towards the hull

The faces that are visible are precisely those that need to be discarded

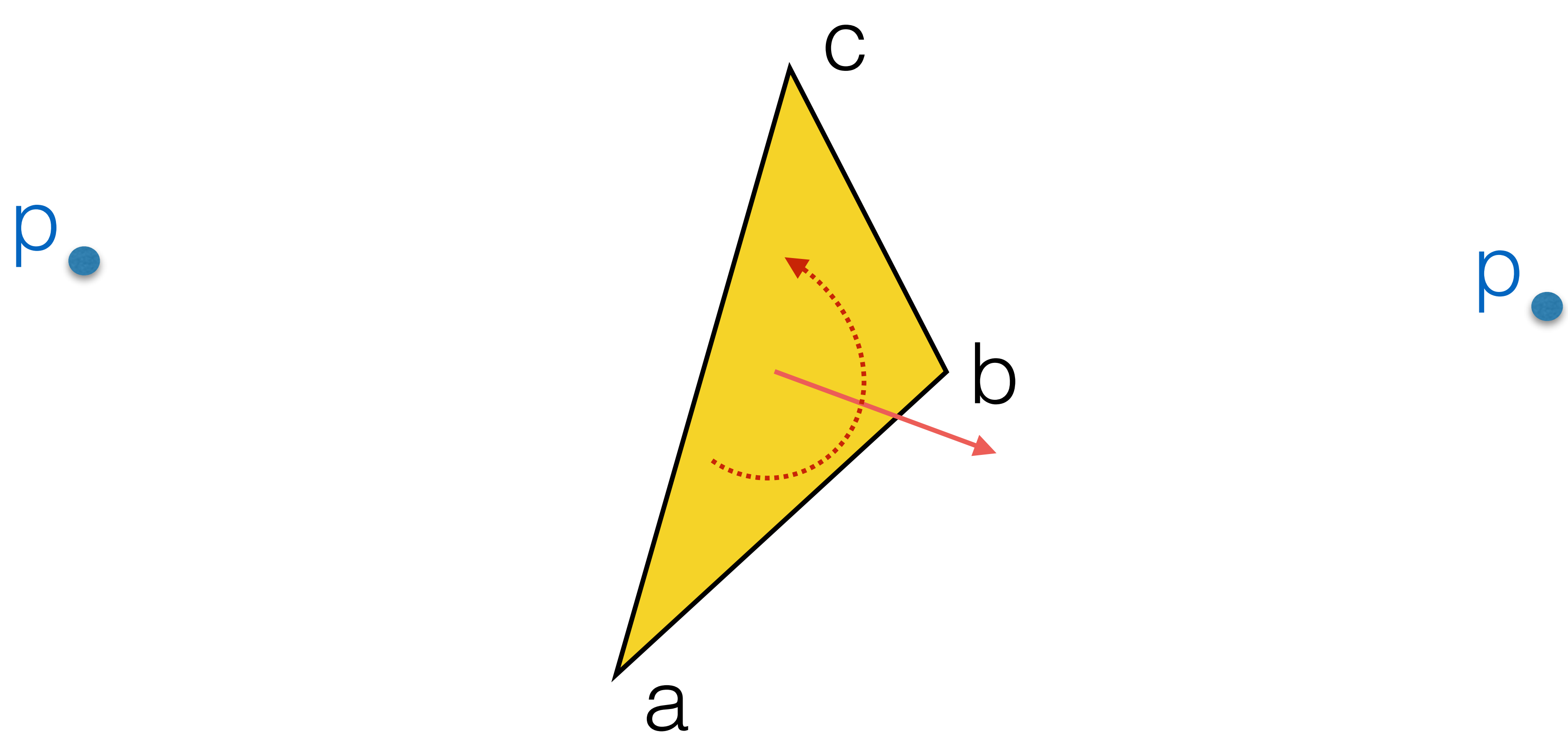
The edges on the border of the visible region become the basis of the cone

Incremental 3d hull

- sort points lexicographically
- initialize H for p1, p2, p3, p4
- for each remaining point p in order
    - for each face f of H: check if f is visible from p
    - if no faces are visible
        - discard p (p must be inside H)
    - else
        - find border edge of all visible faces
        - for each border edge e construct a face (e,p) and add to H
        - for each visible face f: delete f from H

We need a precise definition of visibility

Terminology: Point in front/behind face



ps is left of (behind) abc

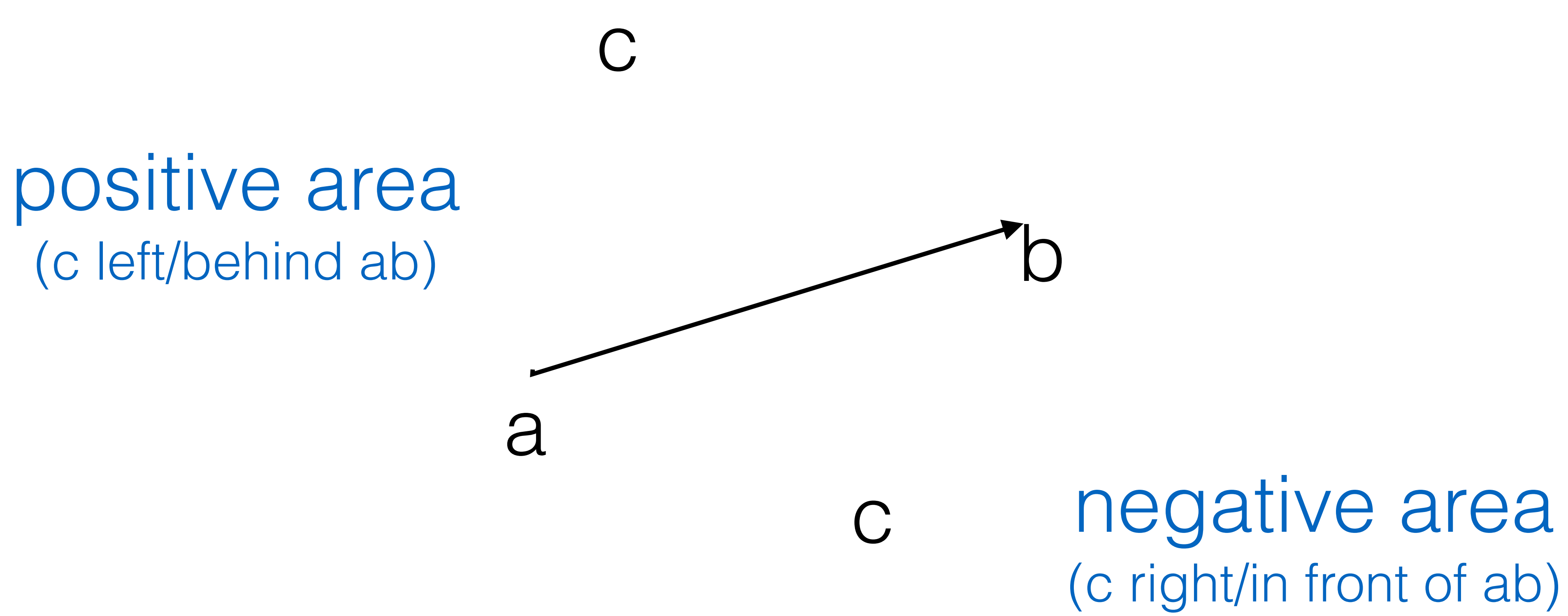
abc not visible from p

p is right of (in front) abc

abc visible from p

2D

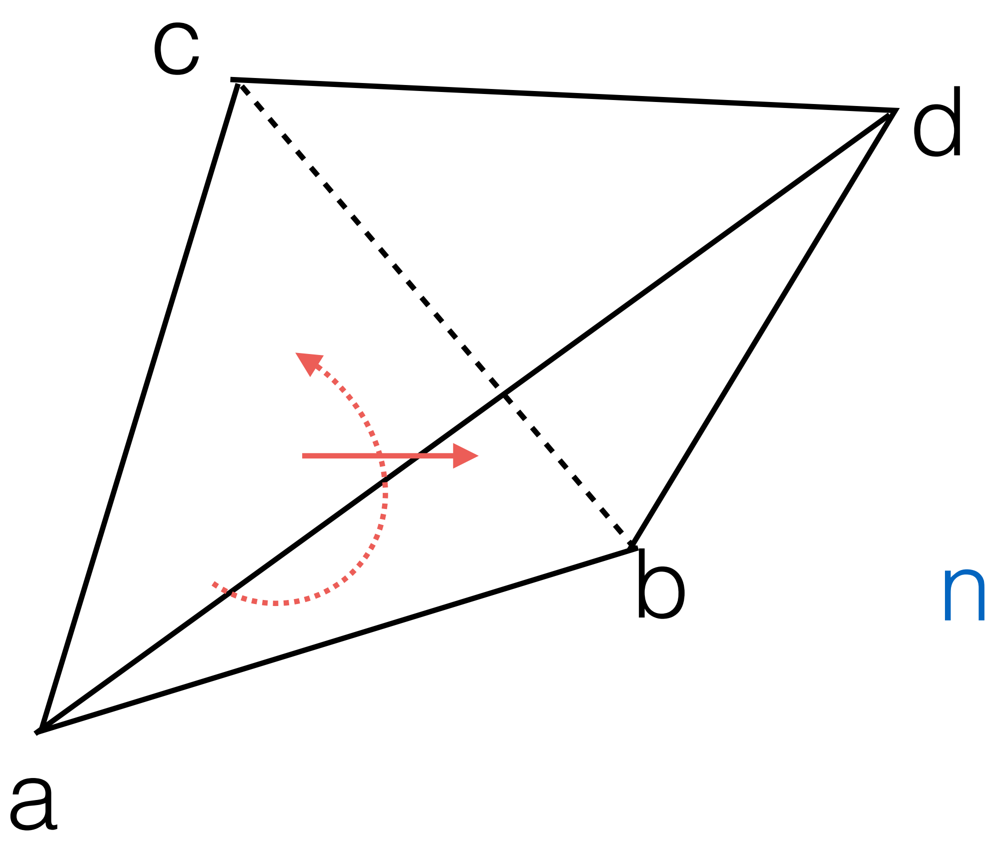
$$2\ \text{signedArea}(a,b,c) = \det \begin{vmatrix} a.x & a.y & 1 \\ b.x & b.y & 1 \\ c.x & c.y & 1 \end{vmatrix}$$



3D

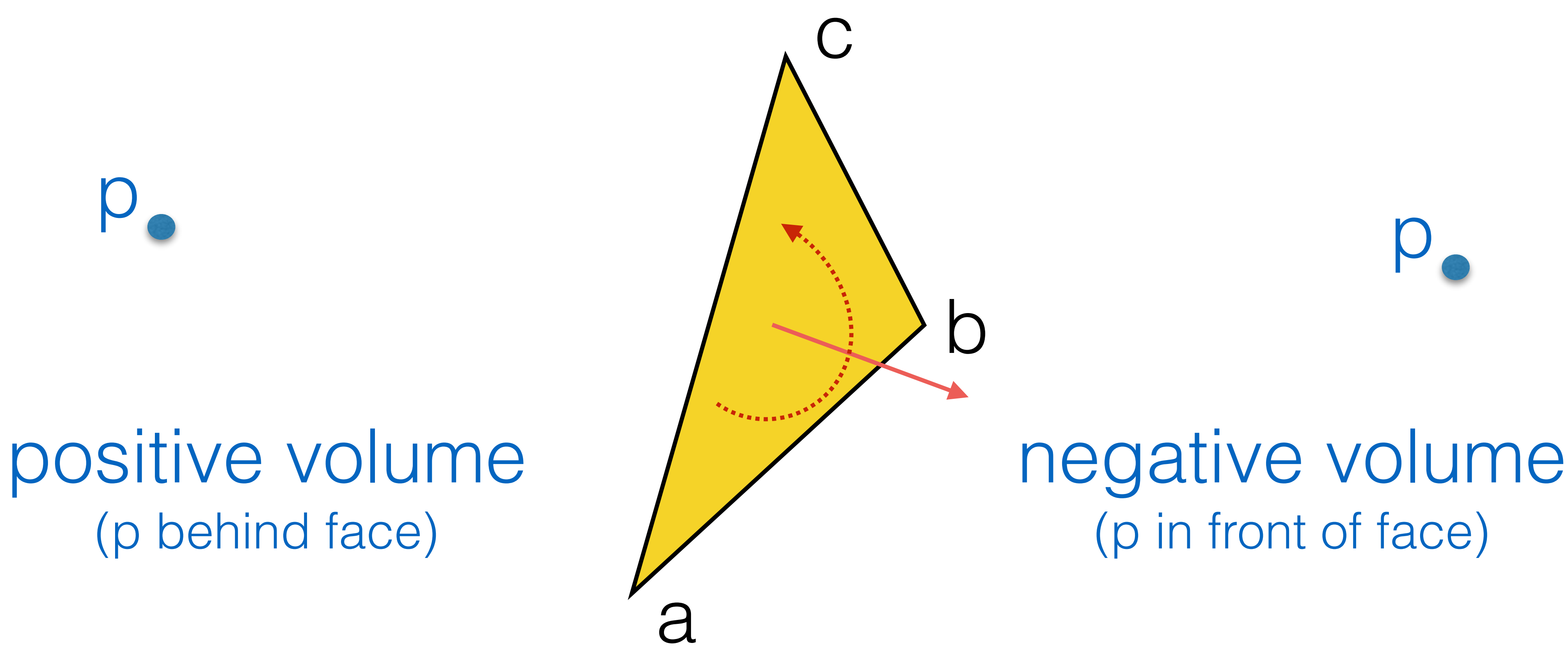
$$6 \text{ signedVolume}(a,b,c,d) = \det \begin{vmatrix} a.x & a.y & a.z & 1 \\ b.x & b.y & b.z & 1 \\ c.x & c.y & c.z & 1 \\ d.x & d.y & d.z & 1 \end{vmatrix}$$

positive volume
(p behind face)



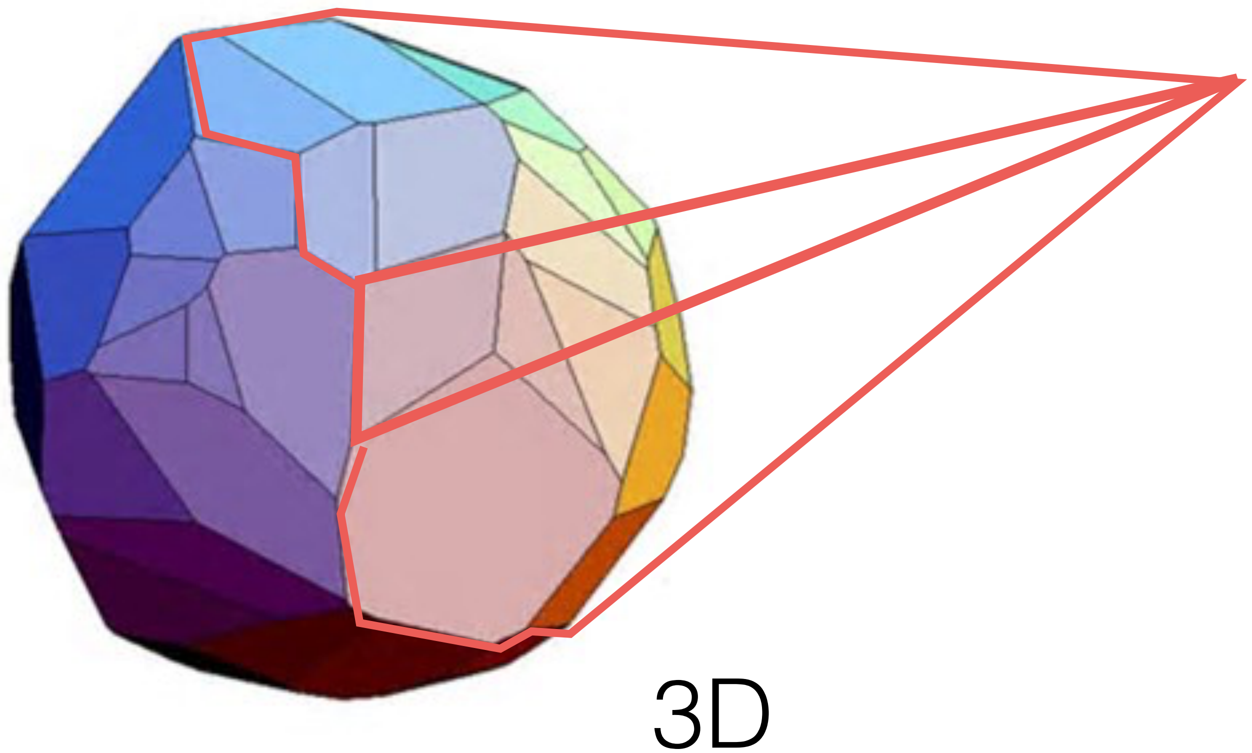
negative volume
(d in front of face)

- Assume all faces oriented counterclockwise so that their normals determined by the right-hand rule point towards the **outside** of P.



is_visible(a,b,c,p): return signedVolume(a,b,c,p) < 0

Incremental 3d hull



The visible faces are precisely those that need to be discarded

The edges on the boundary of the visible region are the basis of the cone

Incremental 3d hull

- Analysis:
  - (Like in 2D) We can start at the previous vertex, find its neighboring faces, determine if they are visible, and continue. For each face that we determine to be visible, that face will be deleted.
  - In 2D: a vertex on the hull is connected to precisely 2 edges. If the vertex is deleted later, deleting the edges can be “charged” to the vertex
  - IN 3D: All faces (e, p) added at step i are now connected to vertex p.   The number of faces incident to a vertex p is not constant and can be $O(n)$. Some or all of these faces may be deleted later.
  - Overall in 3D running time adds up to $O(n^2)$

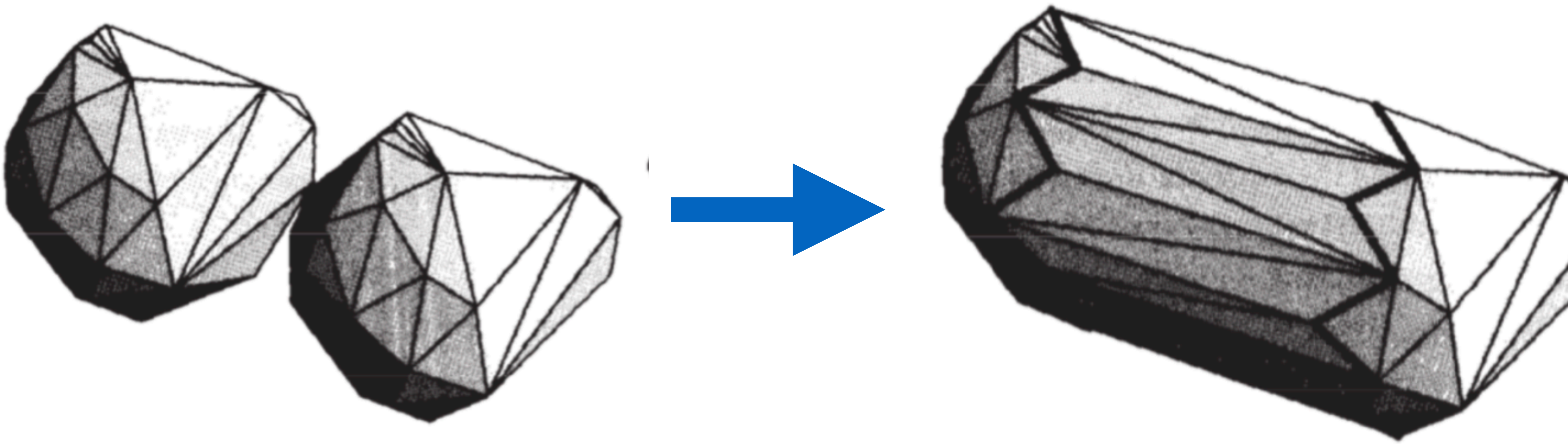
3D hull via divide & conquer

3d hull via divide & conquer

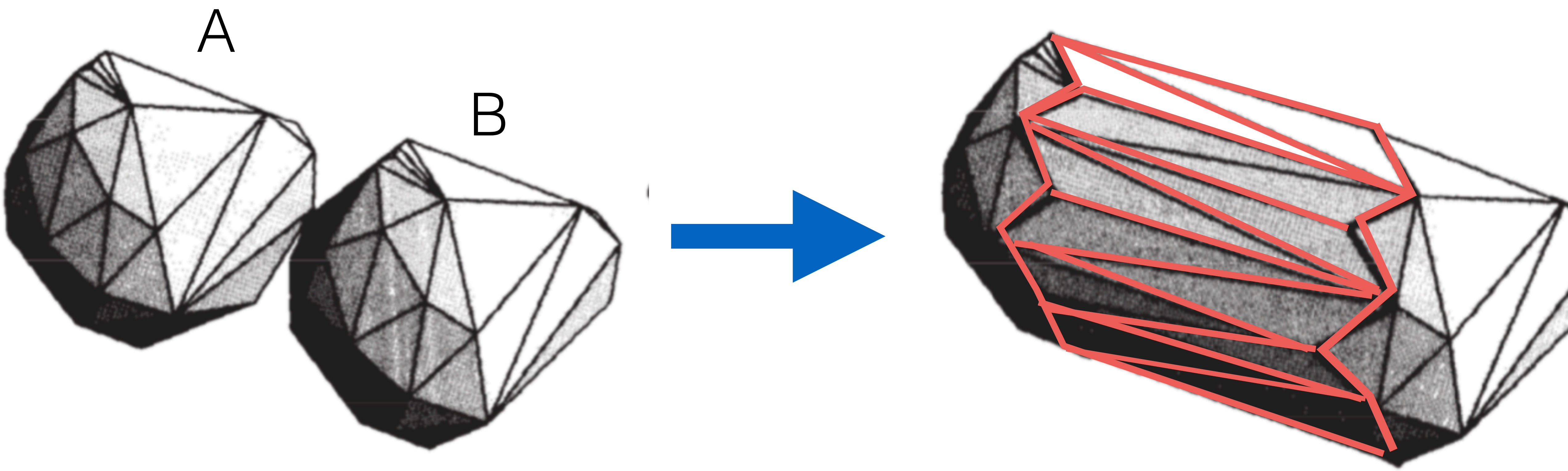
- divide points in two halves P1 and P2
- recursively find CH(P1) and CH(P2)
- merge

- We’ll see that merging can be done in O(n) time ==> O(n lg n) algorithm

Merging

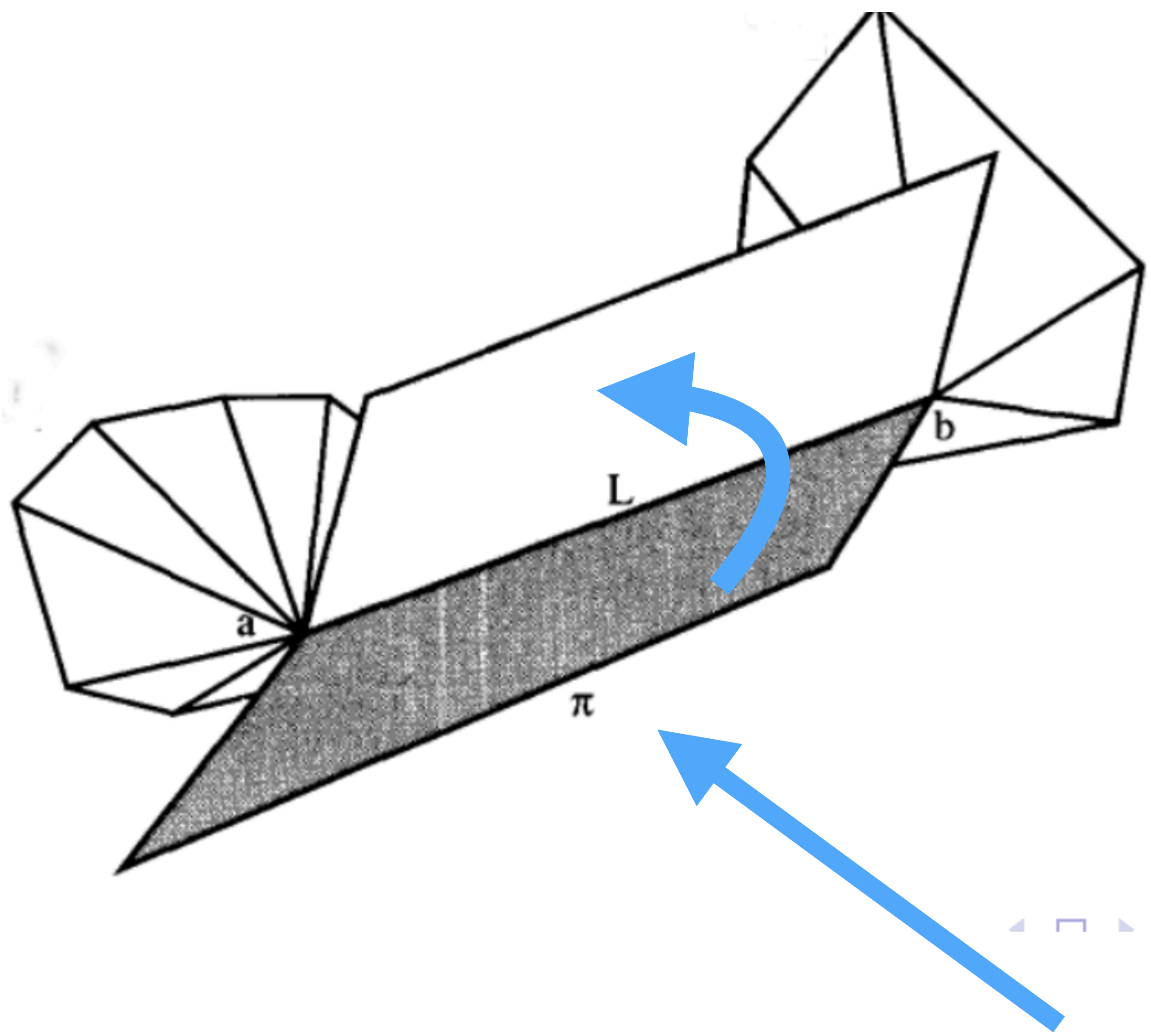


Merging



The merged hull will add a “band” of faces between A and B

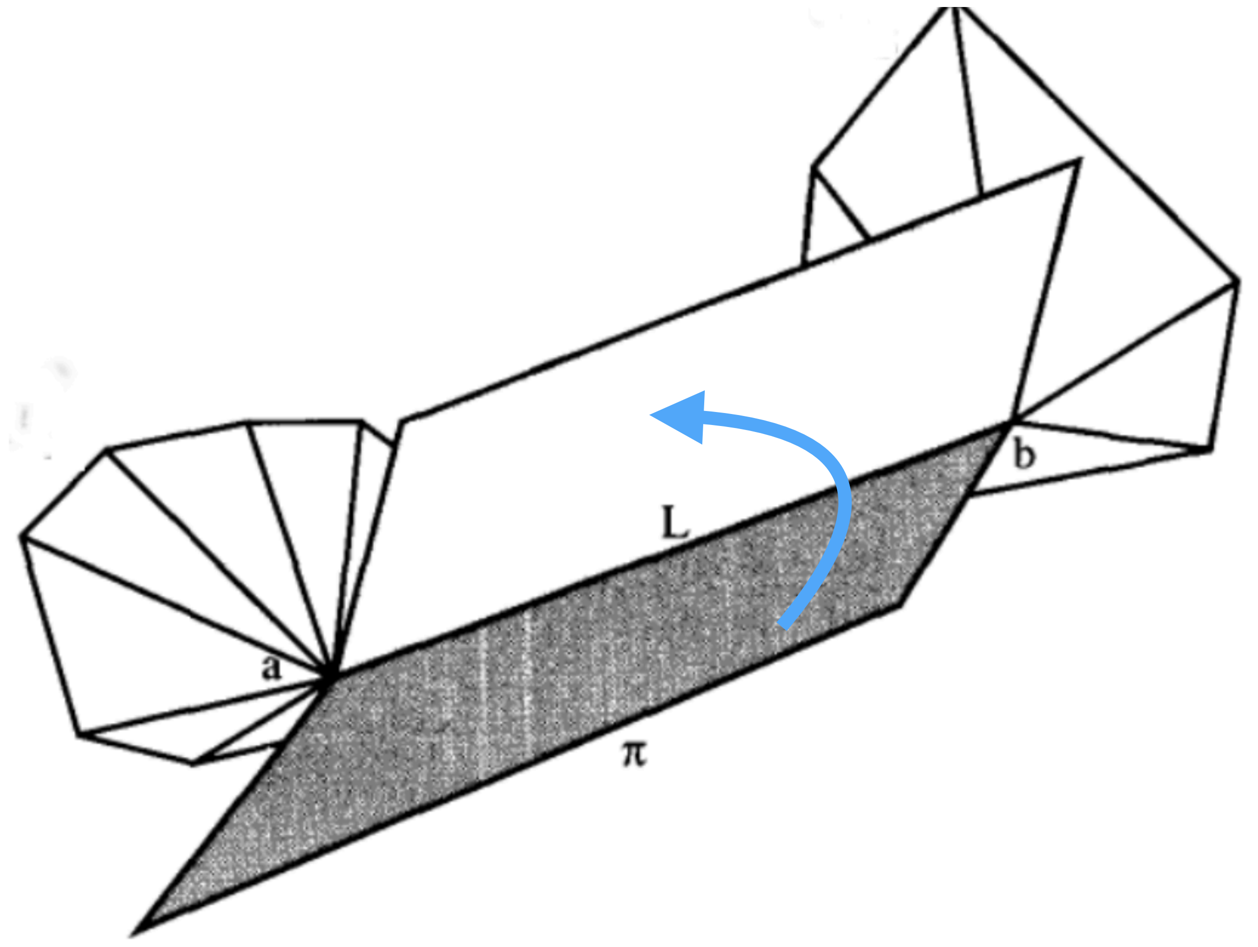
- Imagine rotating the plane around ab, until it touches the polytops A and B



Rotate π around ab

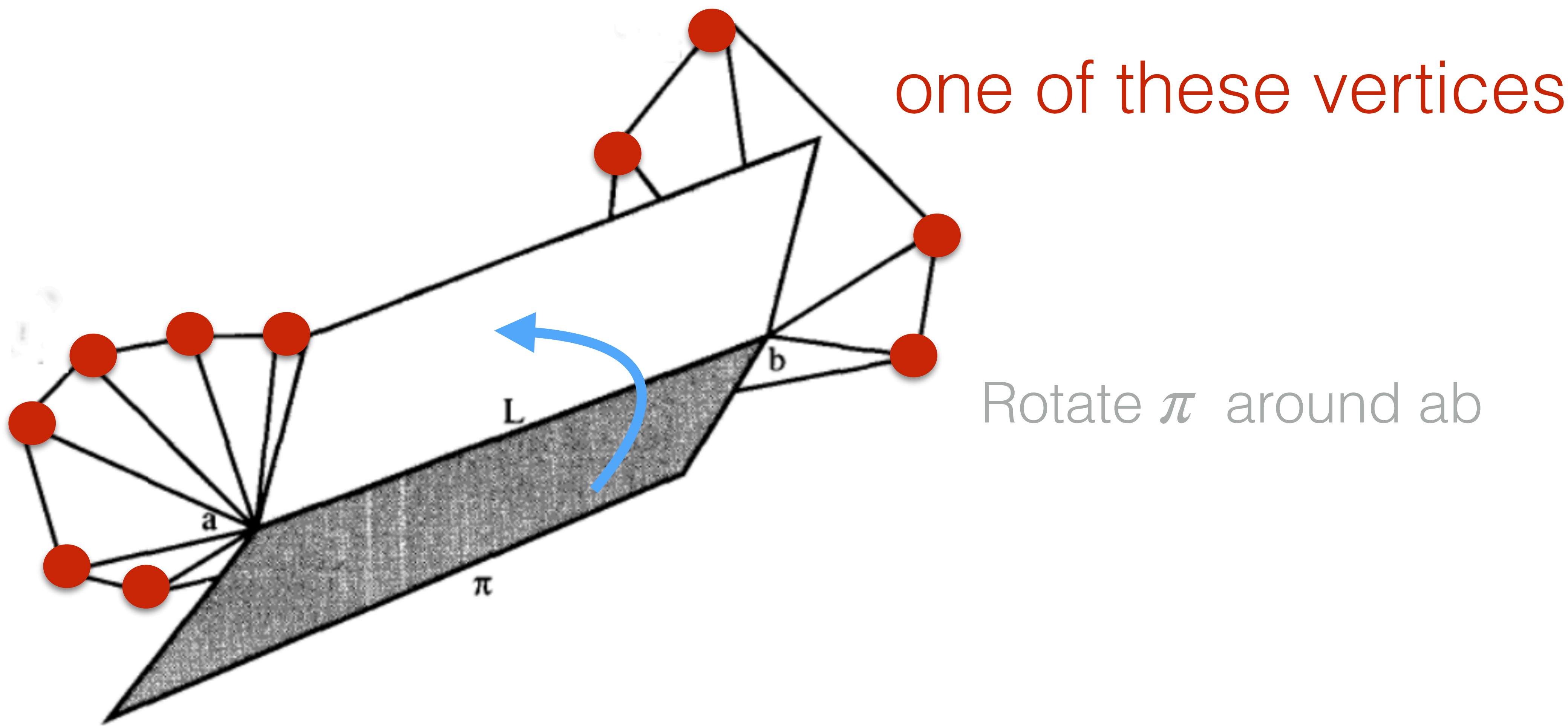
Let π be a plane touching A in a and B in b

- Claim: When we rotate π around ab, the first vertex hit is a vertex c adjacent to a or b and vertex c has the smallest angle among all neighbors of a,b

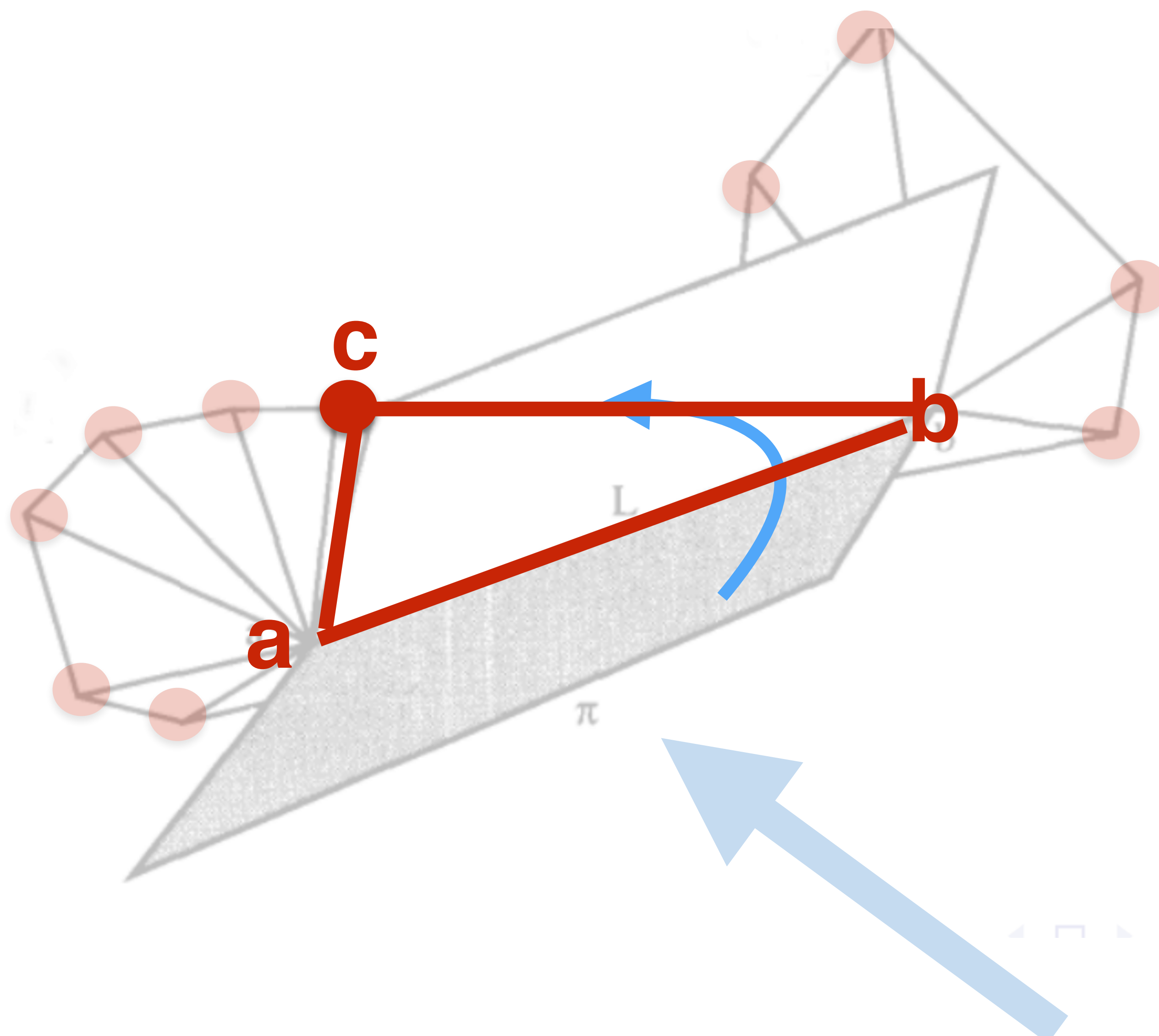


Rotate π around ab

- Claim: When we rotate π around ab, the first vertex hit is a vertex c adjacent to a or b and vertex c has the smallest angle among all neighbors of a,b



- Once π hits c, a triangular face of the merged hull has been found

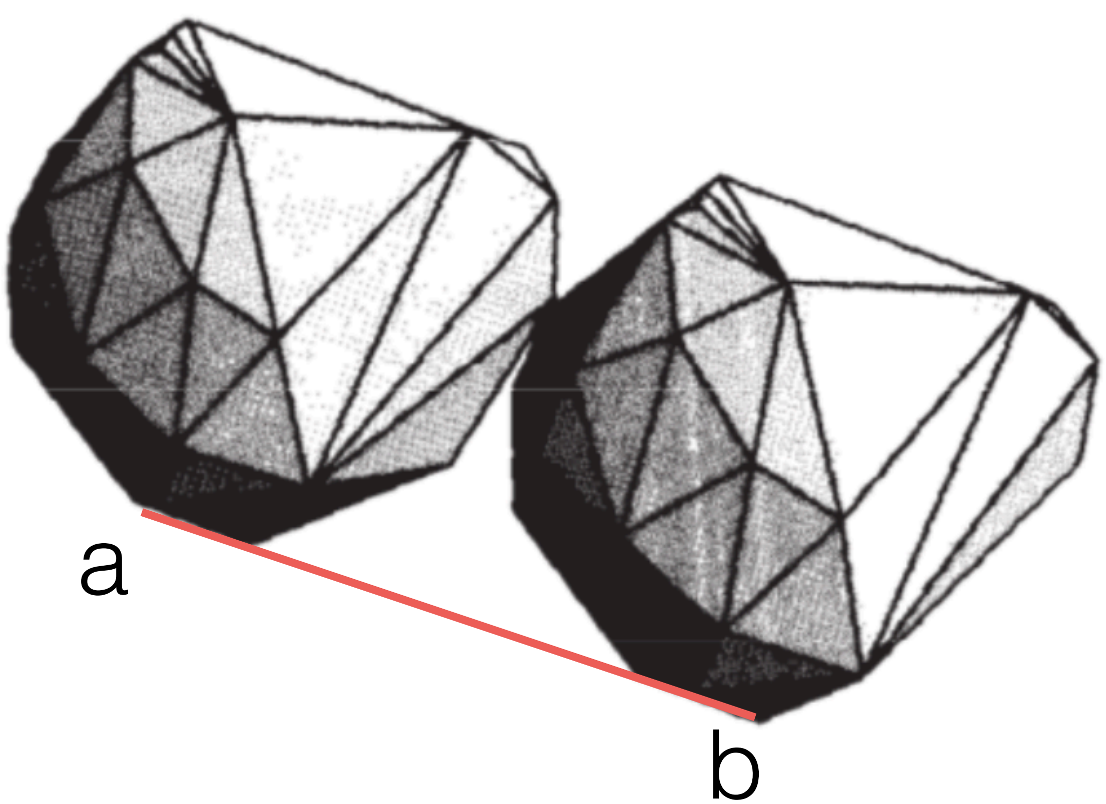


Rotate π around ab

Let π be a plane touching A in a and B in b

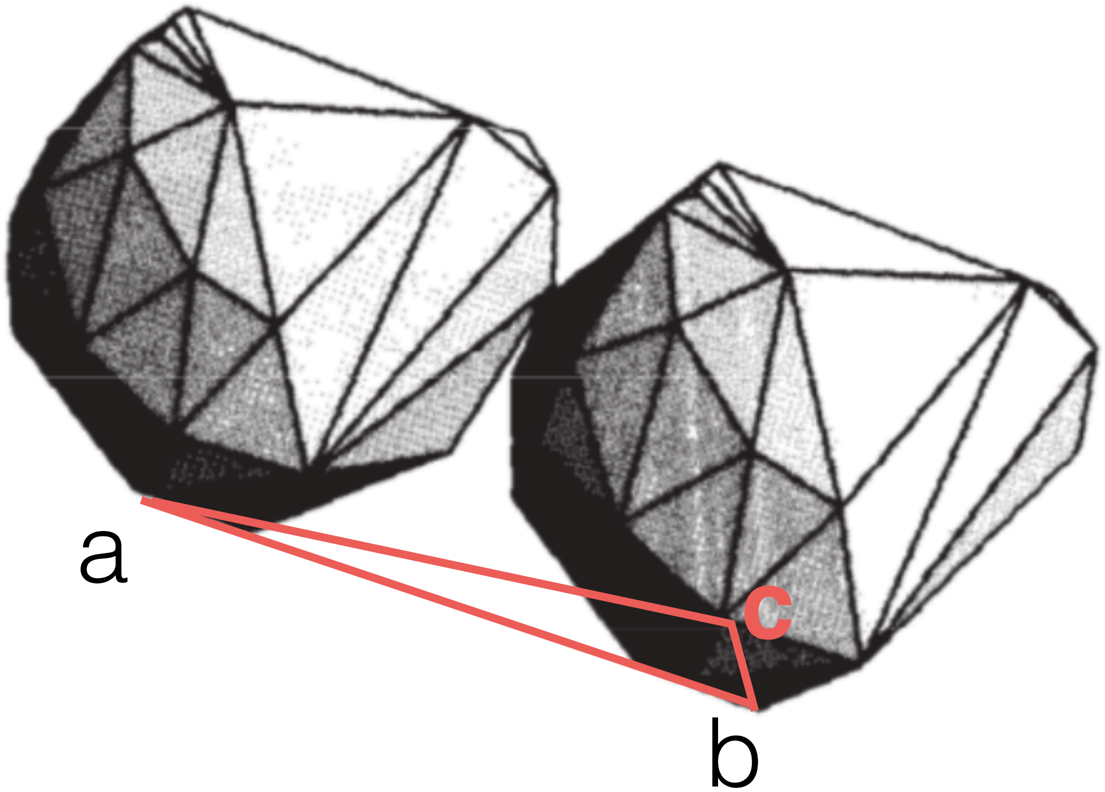
Merge

1. Find a common tangent ab



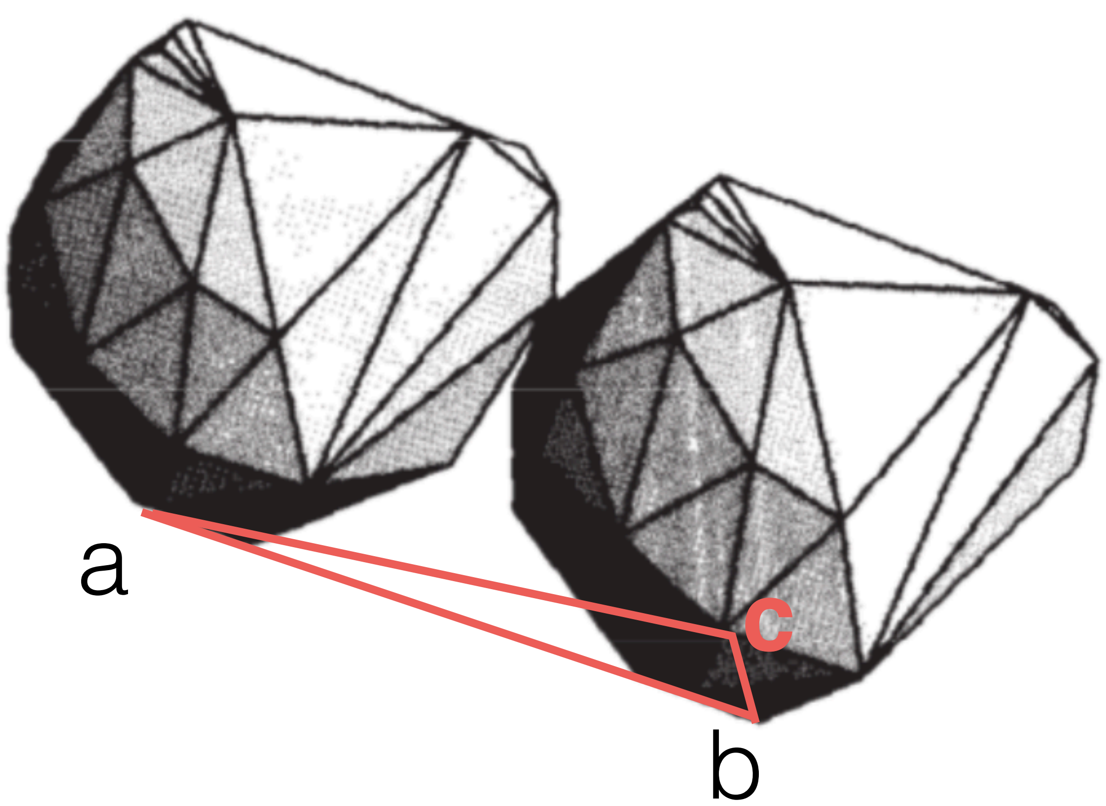
Merge

1. Find a common tangent ab
2. Consider all neighbor vertices of a,b and find the vertex with smallest angle (wrt the plane through ab).



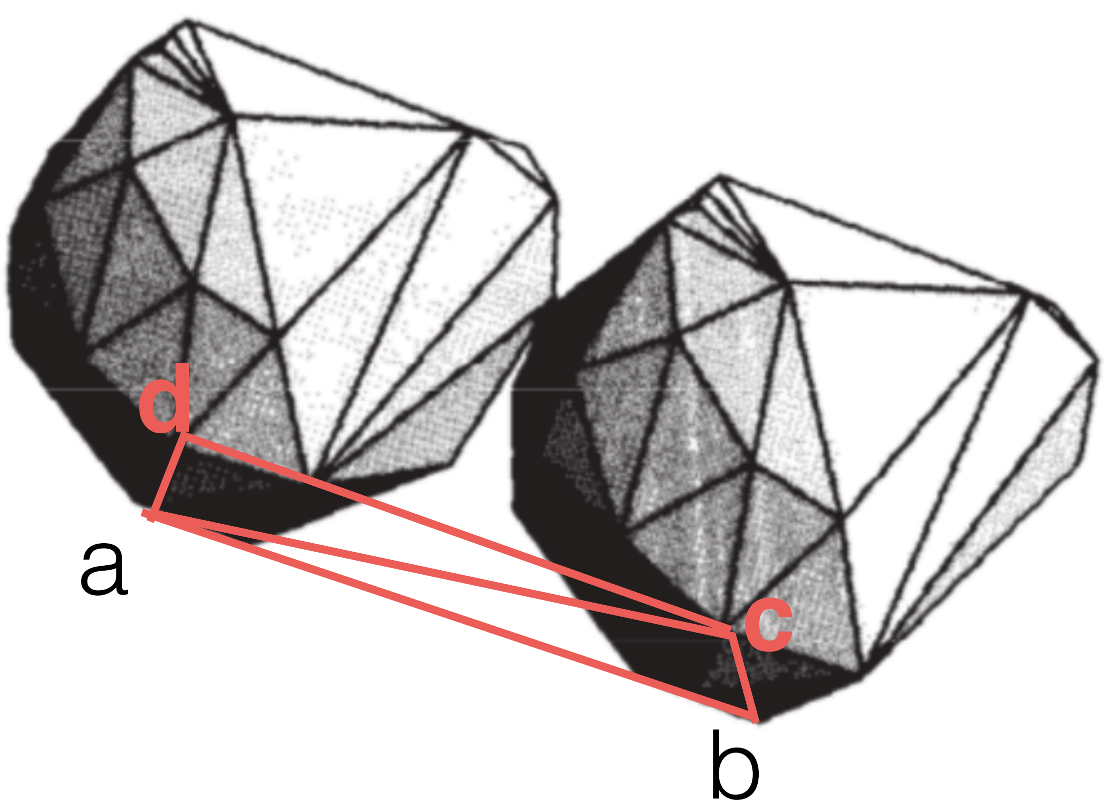
Merge

1. Find a common tangent ab
2. Consider all neighbor vertices of a,b and find the vertex with smallest angle (wrt the plane through ab).
3. Repeat from edge ac.



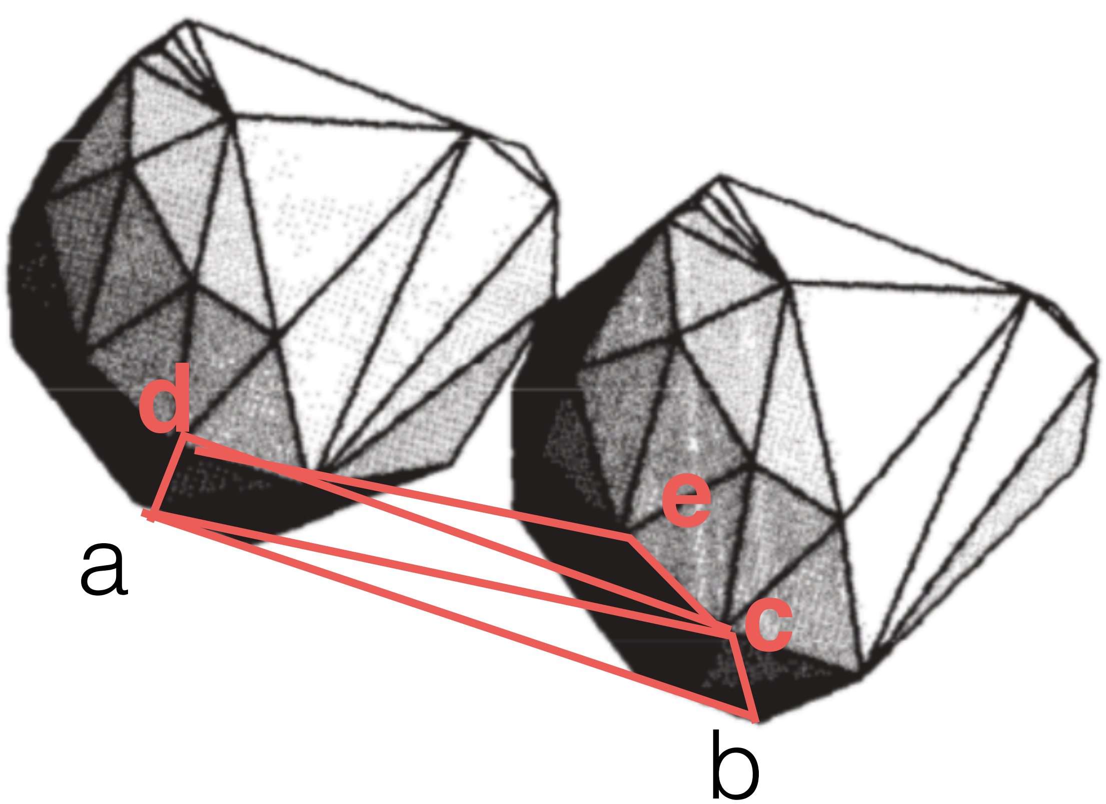
Merge

1. Find a common tangent ab
2. Consider all neighbor vertices of a,b and find the vertex with smallest angle (wrt the plane through ab).
3. Repeat from edge ac.



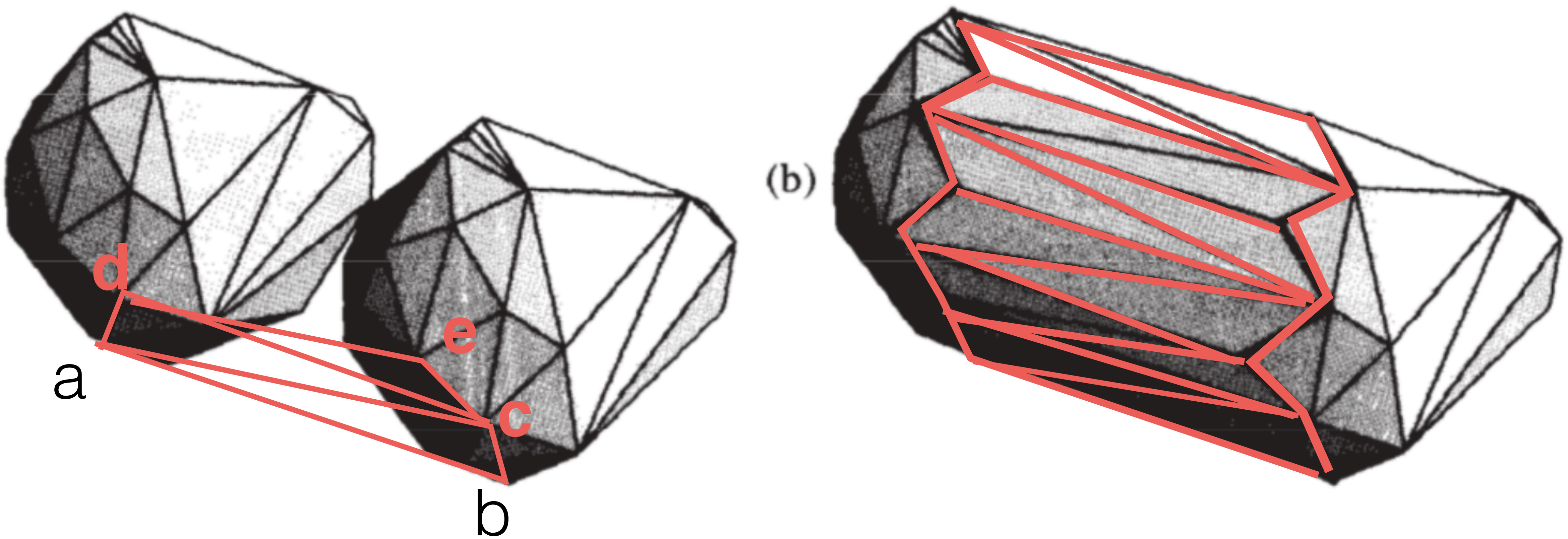
Merge

1. Find a common tangent ab
2. Consider all neighbor vertices of a,b and find the vertex with smallest angle (wrt the plane through ab).
3. Repeat from edge ac.



Merge

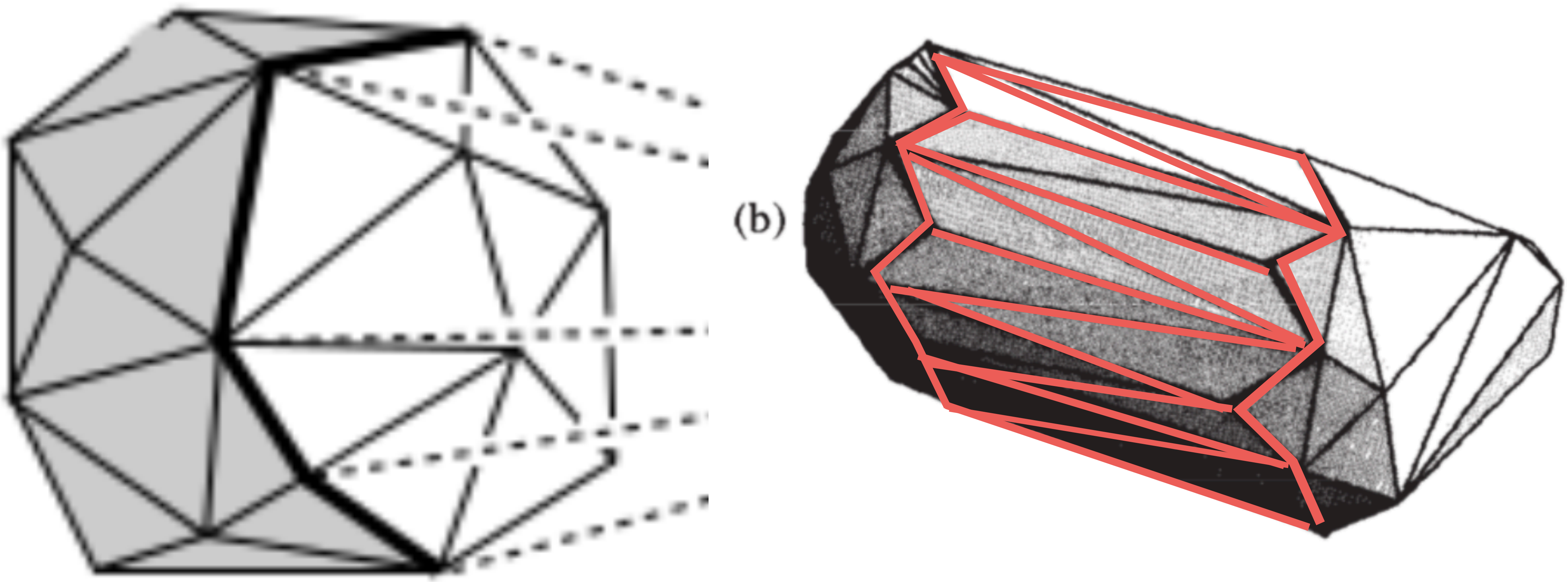
1. Find a common tangent ab
2. Consider all neighbor vertices of a,b and find the vertex with smallest angle (wrt the plane through ab).
3. Repeat from edge ac.



Merge

1. Find a common tangent ab
2. Consider all neighbor vertices of a,b and find the vertex with smallest angle (wrt the plane through ab).
3. Repeat from edge ac.
4. Delete hidden faces

The hidden faces



- Find the edges on the “boundary” of the cylinder
- BFS or DFS faces “towards” the cylinder
- All faces reached are inside

3d hull: summary

3D hull summary

2D		3D
Naive	$O(n^3)$	$O(n^4)$
Gift wrapping	$O(nh)$	$O(n \times F)$
Graham scan	$O(n \lg n)$	does not extend to 3D
Quickhull	$O(n \lg n)$, $O(n^2)$	
Incremental	$O(n \lg n)$	$O(n^2)$
Divide-and-conquer	$O(n \lg n)$	$O(n \lg n)$

3d hull: Summary

- Of all algorithms that extend to 3D, divide-and-conquer is the only one that achieves optimal $O(n \lg n)$
- But, difficult to implement
- The slower algorithms (quickhull, incremental) preferred in practice

Convex hull in higher dimensions

- Surprisingly, have many applications !
    - e.g. computing triangulations for points in 3D can be constructed from convex hulls in 4D
- Size of d-hull: $\Omega(n^{\lfloor d/2 \rfloor})$
- In 4D: size is $\Omega(n^2)$
    - $O(n \lg n)$ algorithm not possible
    - $O(n^2)$ algorithms known

Euler’s formula

- Euler noticed a remarkable regularity in the number of vertices, edges and faces of a polyhedron (w/o holes).

- **Euler’s formula: $V - E + F = 2$**

- Proof idea:
    - flatten the polygon to a plane
    - prove the formula for a tree
    - prove for any planar graph by induction on E