

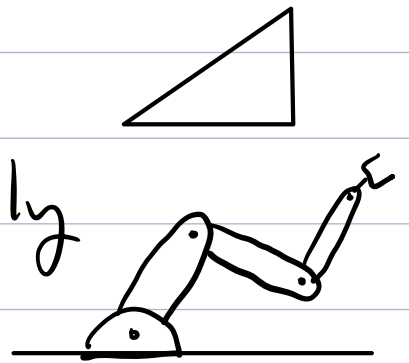
CMSC 754 - Computational Geometry

Lecture 20 - Motion Planning

Motion Planning:

Given a robot (with constraints on how it can move), a set of obstacles, and a start + target configurations for the robot, is there a collision-free motion plan?

Robot: May be rigid object or linked/hinged assembly



Motion constraints:

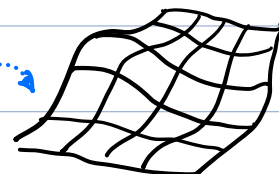
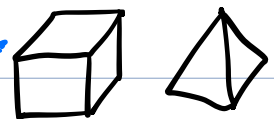
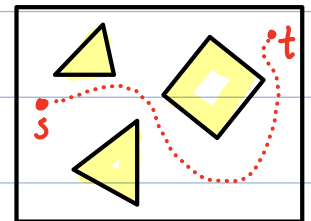
- Translation
- Rotation
- Speed/Acceleration limits
- ⋮

Obstacles: Polygons in 2-D

Polyhedra in 3-D

Curved objects

Terrains



We'll mostly consider the simplest scenario:

Space - \mathbb{R}^2

Robot - (convex) polygon

Motion - translation only

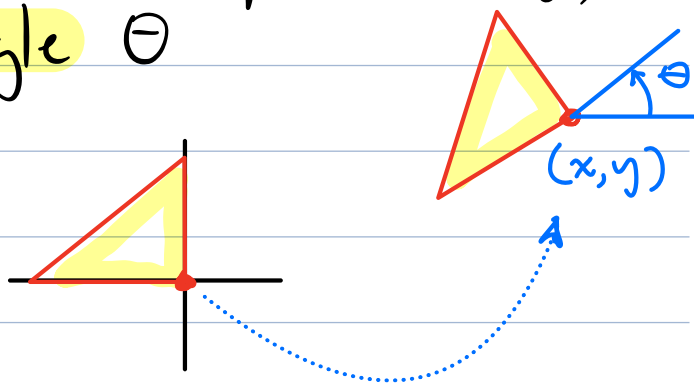
Obstacles - collection of nonoverlapping convex polygons

Configuration: A set of parameters that uniquely specifies the robot's position

E.g. Rigid in 2-D

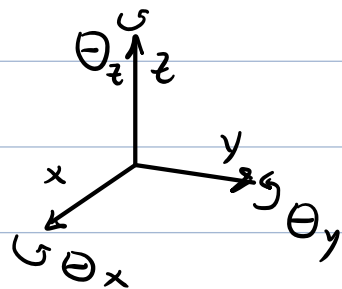
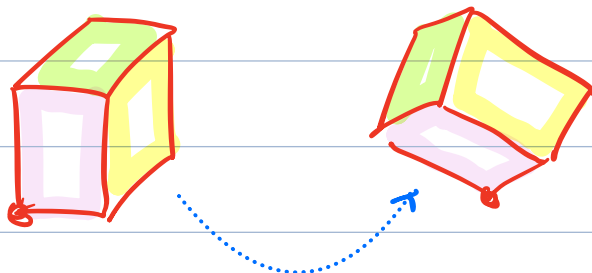
- location of reference point (x, y)
- rotation angle θ

Reference position:

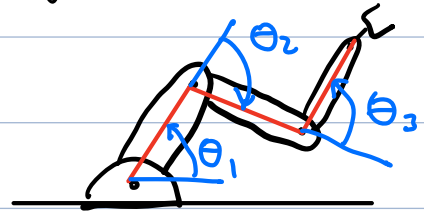


Rigid in 3-D

- location (x, y, z)
- rotation
 - Euler angles $(\theta_x, \theta_y, \theta_z)$
 - Quaternion



Linked/Hinged: Joint angles
($\theta_1, \theta_2, \theta_3$)



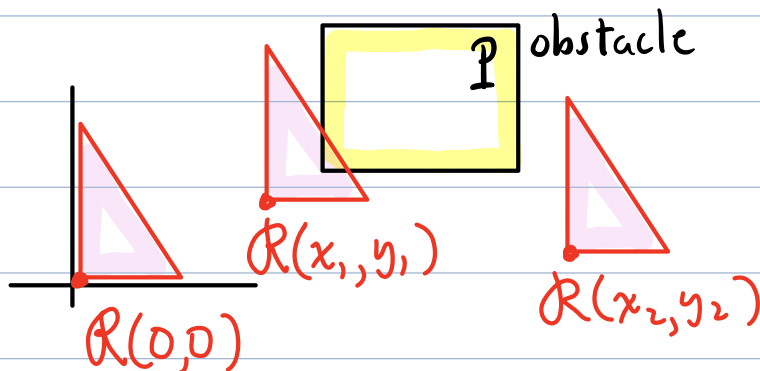
Motion Planning in Config. Space:

- Rather than moving a robot amidst obstacles
- instead -
- Move a point in the robot's configuration space

Need to distinguish between:

free configuration - robot does not collide
forbidden configuration - robot collides

E.g. Translation only (configuration = location of ref. point)



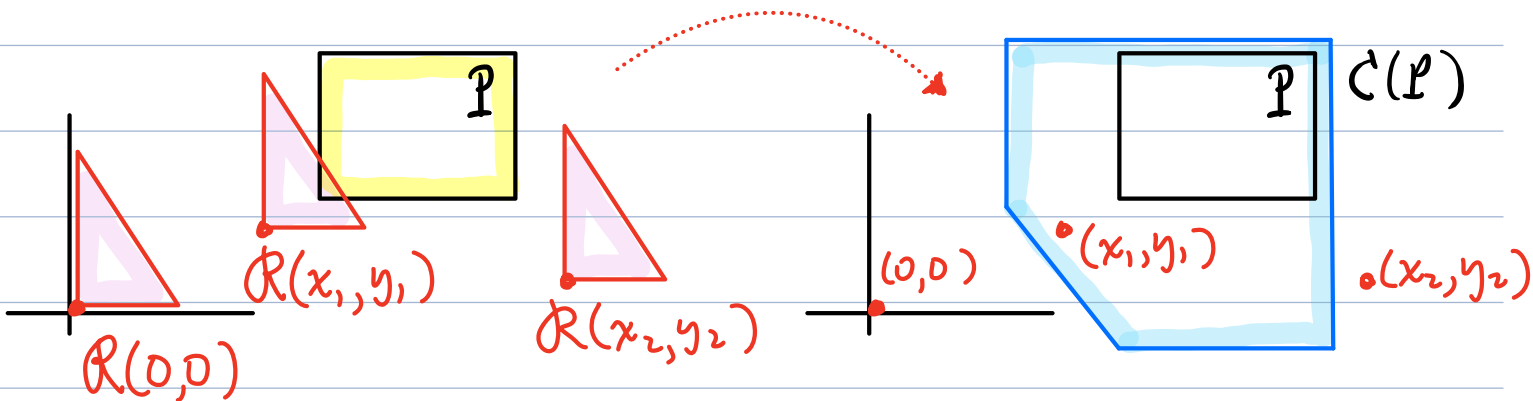
(x_1, y_1) - forbidden
 $R(x_1, y_1) \cap P \neq \emptyset$

(x_2, y_2) - free
 $R(x_2, y_2) \cap P = \emptyset$

Configuration Obstacle (or C-Obstacle)

Given robot R , config vector v , obstacle P
the C-obstacle for P is:

$$C(P) = \{v \mid R(v) \cap P \neq \emptyset\}$$



C-Obstacles for Translation - Minkowski Sum

The easiest C-obstacles are for translational motion.

Def: Given $P, Q, S \subseteq \mathbb{R}^d$ + $\alpha \in \mathbb{R}$

$$P \oplus Q = \{p + q : p \in P, q \in Q\}$$

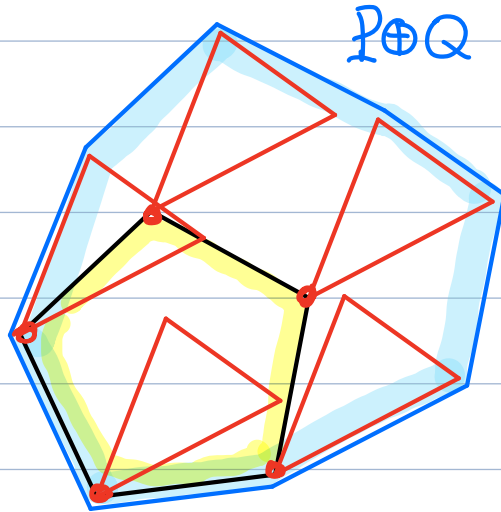
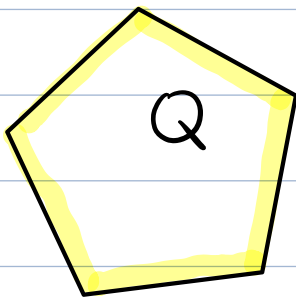
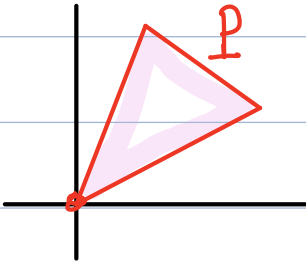
Minkowski Sum

$$\alpha P = \{\alpha \cdot p : p \in P\}$$

$$-P = \{-p : p \in P\}$$

Intuition: $P \oplus Q$ - Place P so its ref. pt. is at origin

- Sweep P 's ref pt around Q
+ see what's swept out



Lemma: Given a translating robot R + obstacle P :

$$C_r(P) = P \oplus (-R)$$

Proof: For any translation vector t

$$t \in C(P) \Leftrightarrow R(t) \text{ collides with } P$$

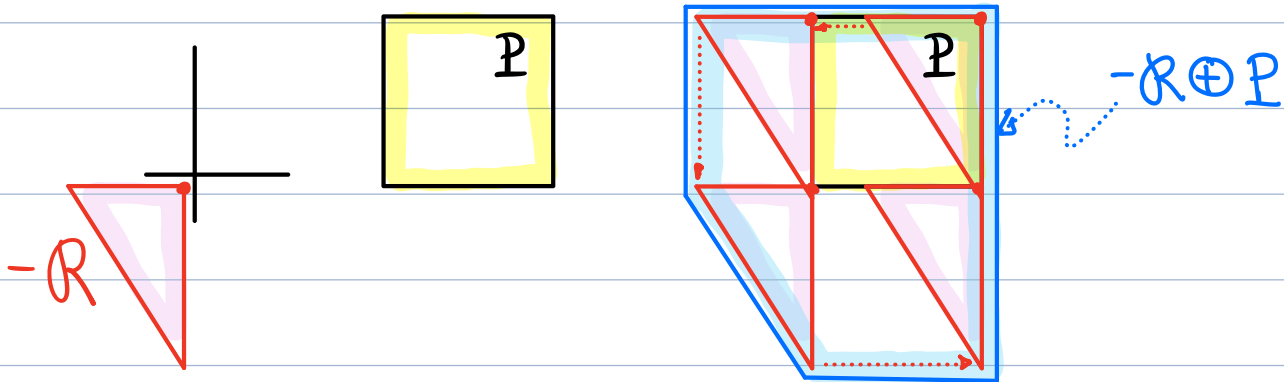
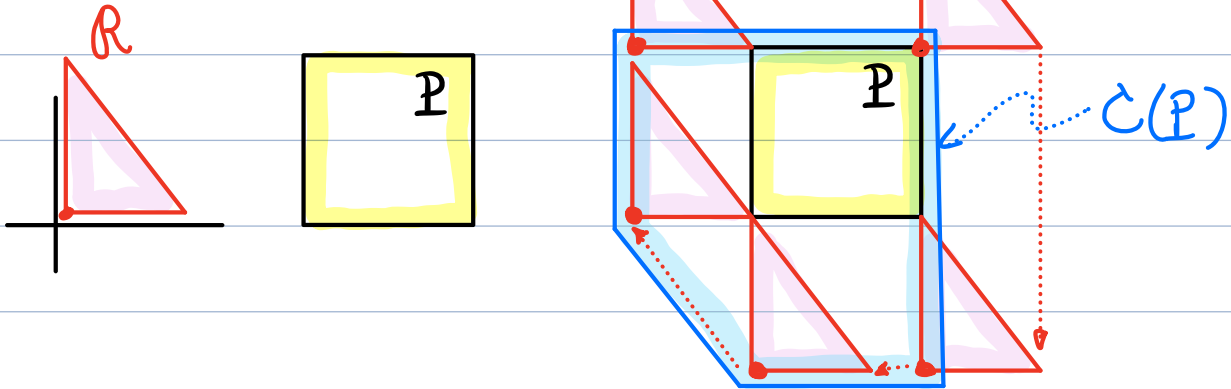
$$\Leftrightarrow R + t \cap P \neq \emptyset$$

$$\Leftrightarrow \exists r \in R, p \in P \quad r + t = p$$

$$\Leftrightarrow \quad \quad \quad t = p - r$$

$$\Leftrightarrow t \in P \oplus (-R)$$

Proof by picture:



Computing the Minkowski Sum:

If P is a convex m -gon

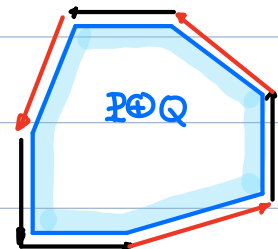
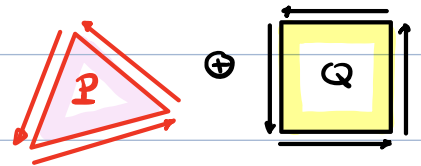
Q is a convex n -gon

can compute $P \oplus Q$ in time $O(m+n)$

- Direct edges (CW) (vectors)

- Sort them by angle

- Join them tail to head



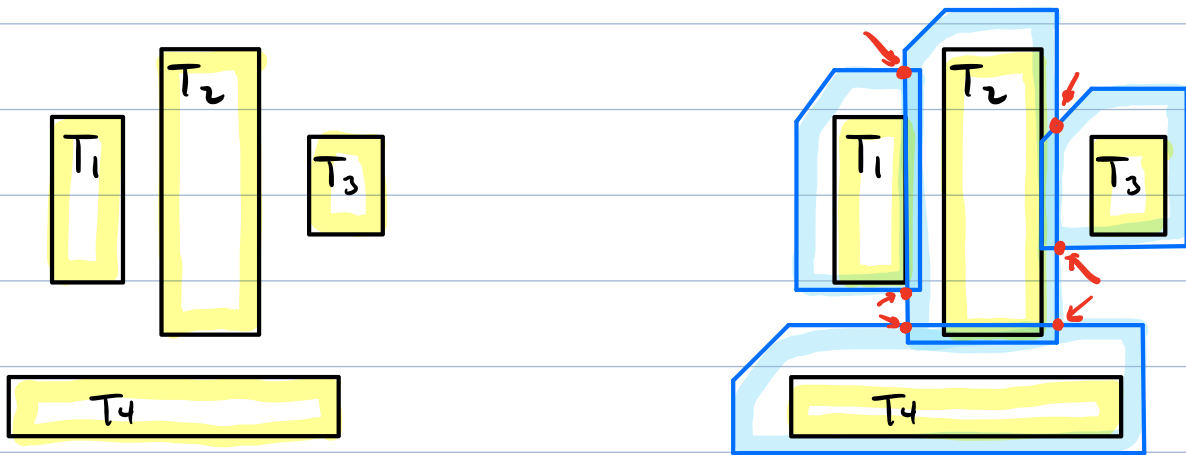
Complexity of C-Obstacles:

- Suppose we have an m -sided convex robot R and a collection of disjoint convex obstacles T_1, \dots, T_k . Let $n_i = \text{num. of sides in } T_i$. Let $n = \sum n_i$.

- What is total size of config. obstacles?

$$\bigcup_{i=1}^k C_R(T_i) = \bigcup_{i=1}^k T_i \oplus (-R)$$

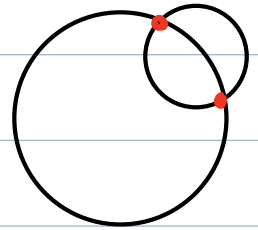
- Although T_i 's are disjoint, $C_R(T_i)$ may overlap



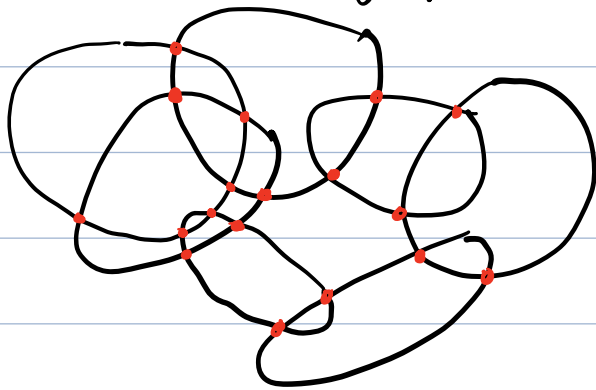
- Points of boundary overlaps create additional vertices - How many? $O(n)$ $O(n^2)$?

Pseudodisks:

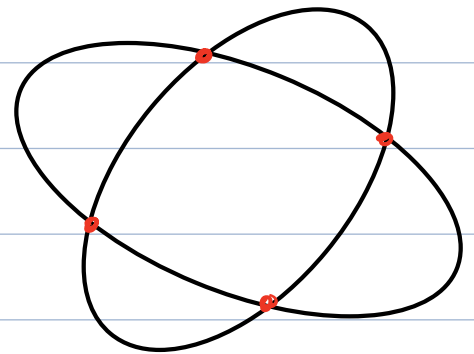
- The boundaries of two circular disks intersect at most twice.



- A collection of convex objects $\{O_1, \dots, O_k\}$ is a collection of pseudodisks if the boundaries of any pair intersect at most twice.



Collection of pseudodisks



Not pseudodisks

Lemma: Given a set T_1, \dots, T_k of disjoint convex bodies in \mathbb{R}^2 and convex R

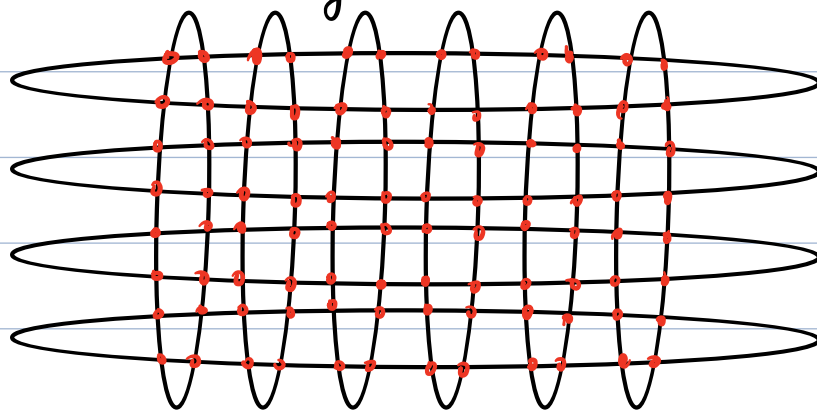
$$\{C_R(T_1), \dots, C_R(T_k)\} \equiv \{T_1 \oplus (-R), \dots, T_k \oplus (-R)\}$$

is a collection of pseudodisks.

Proof: See latex notes

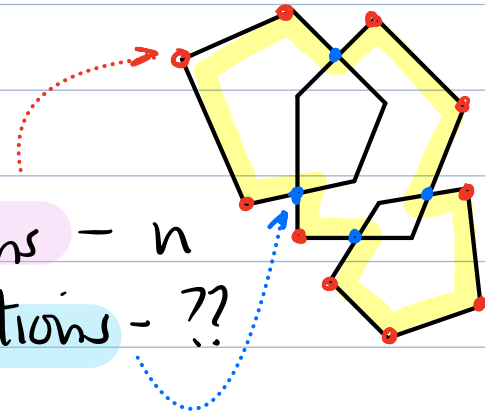
Theorem: Given a collection of pseudodisks with a total of n vertices, their union has a total of $O(n)$ vertices.

In general, union may have $O(n^2)$ vertices



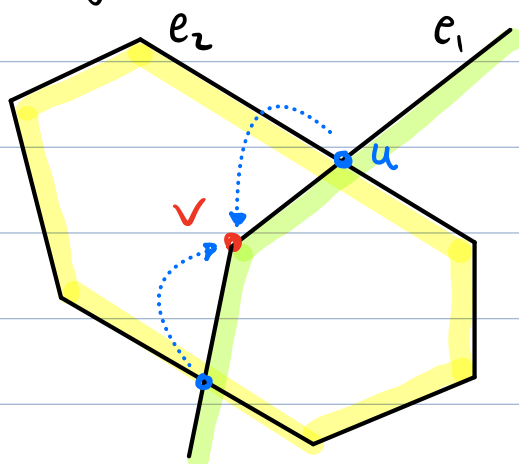
Vertex types:

- Vertices of original polygons - n
- Vertices caused by intersections - ??



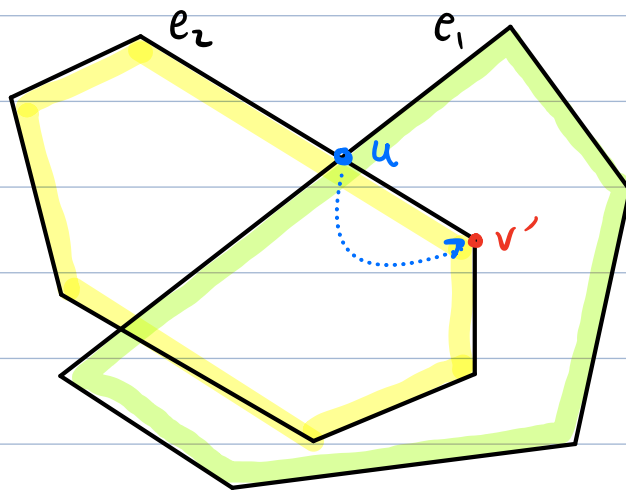
- We'll "charge" intersection vertices to vertices hidden in the interior

- Suppose edges e_1 + e_2 intersect at u



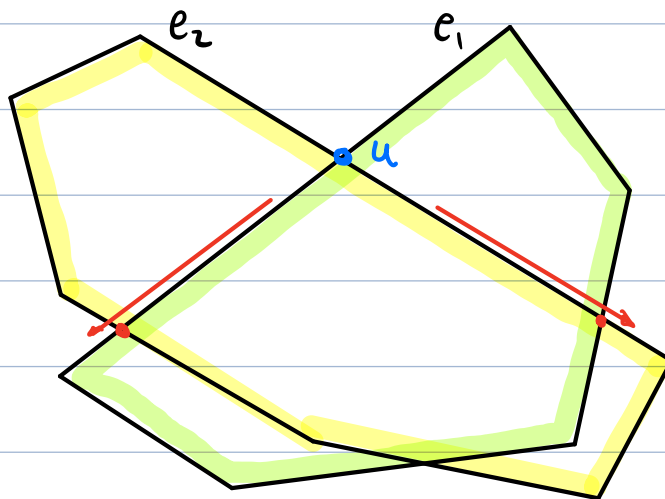
- if e_1 leads to internal vertex v , charge u to v
- v gets ≤ 2 charges

- Otherwise, if e_1 cuts through, but e_2 leads to internal vertex v' , charge u to v'



(Again v' can be charged at most twice)

- Otherwise both $e_1 + e_2$ cut through the other polygon



But this cannot happen since these are pseudodisks!

Since every vertex is charged at most twice union has at most $2n$ vertices. \square

Theorem: Given a convex m -sided robot and a collection of n disjoint obstacles, each with $O(1)$ sides, the total boundary complexity of the union of C -obstacles is $O(m \cdot n)$.

Proof: We have a collection of n pseudodisks each with $O(1) + m = O(m)$ sides.

\Rightarrow Total vertices is $O(m \cdot n)$

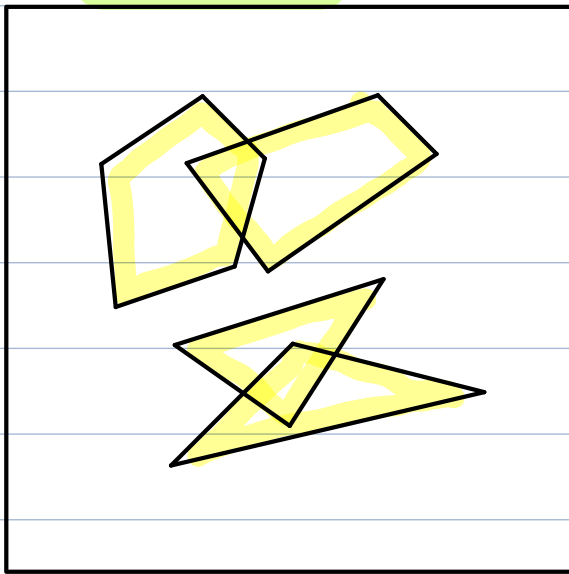
\Rightarrow Union complexity is $O(m \cdot n)$.

Path Planning in Config Space:

Once we have computed the C -obstacles, how to find a path between start + target?

- Compute union of C -obstacles
- Compute a decomposition of the complement space (outside the C -obstacles)
Eg. Triangulate or trapezoid map
- Compute dual graph, joining pairs that can reach each other

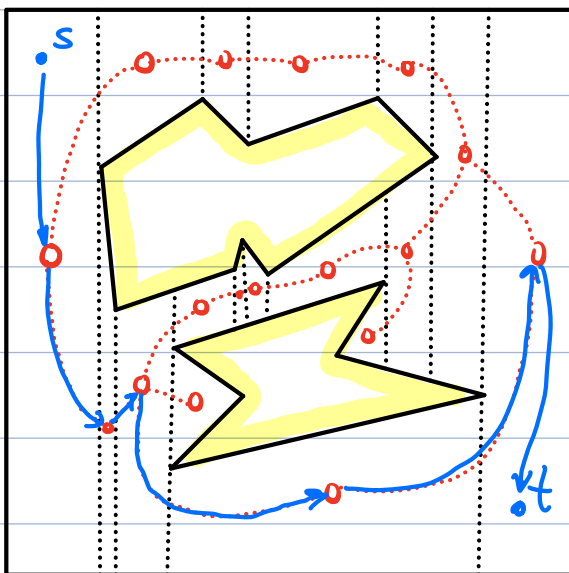
C-obst.



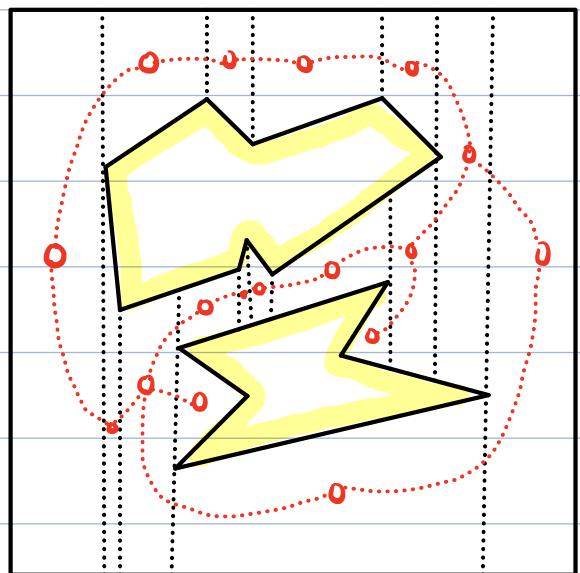
Union



Path s to t



Trap. Map



Finally: Given start s + target t ,

- find trapezoids containing them
- if reachable in dual graph
 - create path joining them
- else - output "unreachable"

Note: Not the shortest path

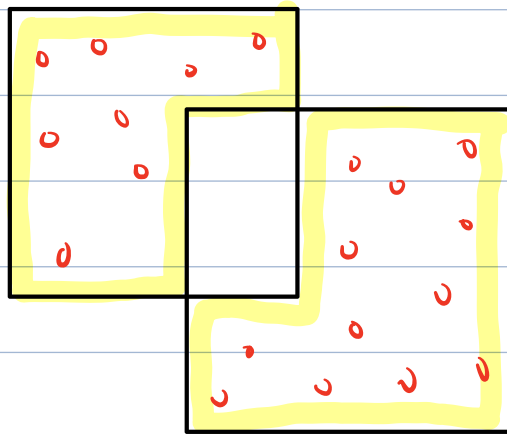
Hints on Bounding VC-Dimension

- Finding upper bounds on complex range spaces is difficult
- Here's a (relatively) easy method to compute (crude) upper bounds

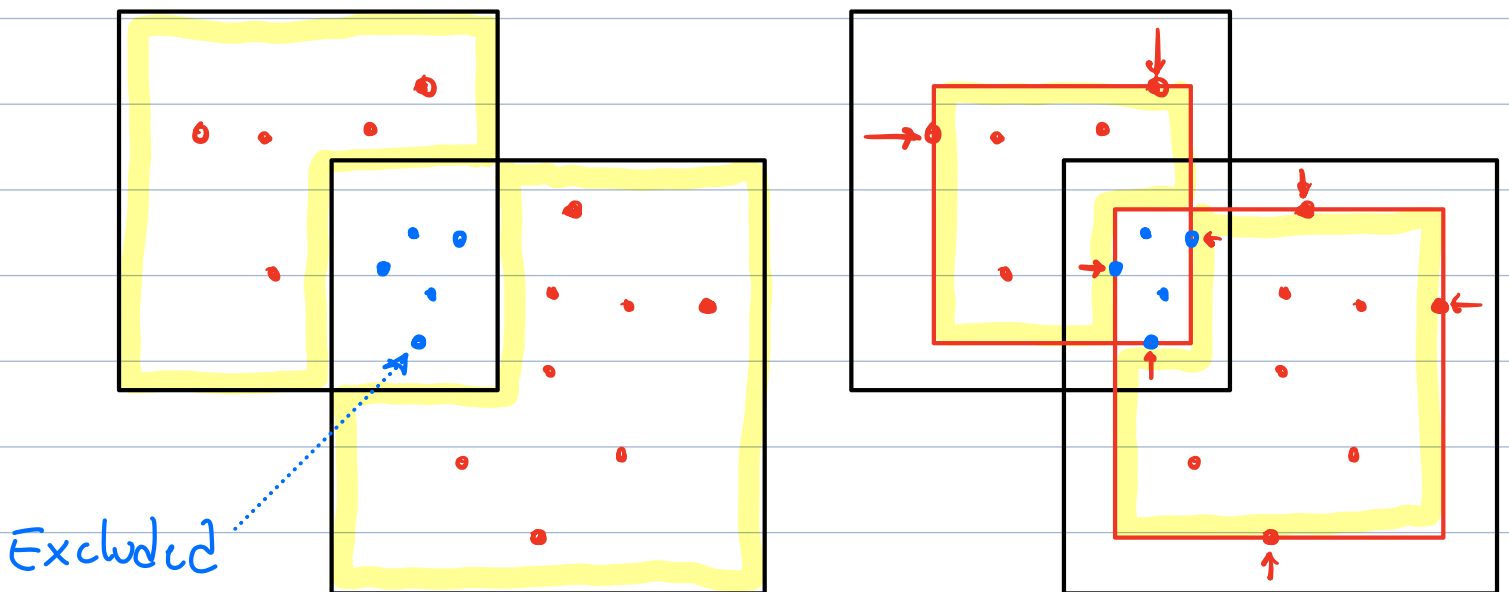
Canonical Shapes:

- Given a geometrically defined range space, construct a mapping from all equivalent shapes (same set of pts) to a single canonical shape
- Count the number of canonical shapes as function of number of pts: $f(n)$
- Be careful to include special cases (eg., empty set)

E.g. $R =$ Symmetric difference of two axis parallel rectangles



Canonical shape: Shrink each rectangle as much as possible with changing the points covered



- Canonical shape defined by ≤ 8 points $\Rightarrow f(n) \leq n^8$

- Typically $f(n)$ is polynomial in n ,
because each canonical shape
depends on $O(1)$ pts.

- Find smallest n s.t.

$$f(n) < 2^n$$

- (Clearly, not enough distinct
ranges to shatter any
 n -element point set.

$$\Rightarrow \text{VC-Dim} : < n$$