CS586:
Configuration Space I

Sung-Eui Yoon
(윤성의)

Course URL:
http://sgvr.kaist.ac.kr/~sungeui/MPA

KAIST

Class Objectives (Ch. 3)

- Configuration space
  - Definitions and examples
  - Obstacles
  - Paths
  - Metrics
- Last time:
  - Classic motion planning approaches including roadmap, cell decomposition and potential field

KAIST

Questions

- **Why is the voronoi diagram algorithm in O(n log n) time, how does the algorithm achieve this and how does it work.**

Voronoi-Based Multi-Robot Autonomous Exploration in Unknown Environments via Deep Reinforcement Learning

- **Use Voronoi regions to reduce duplicated explorations**
- **Published at IEEE T. on Vehicular Tech., 2020**
- **500 cited**

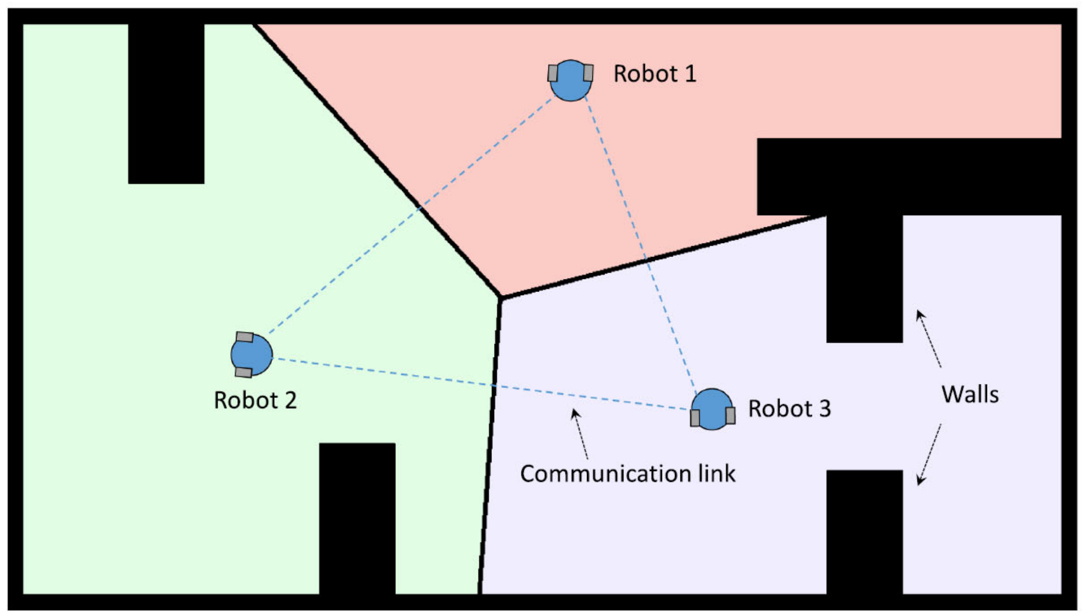
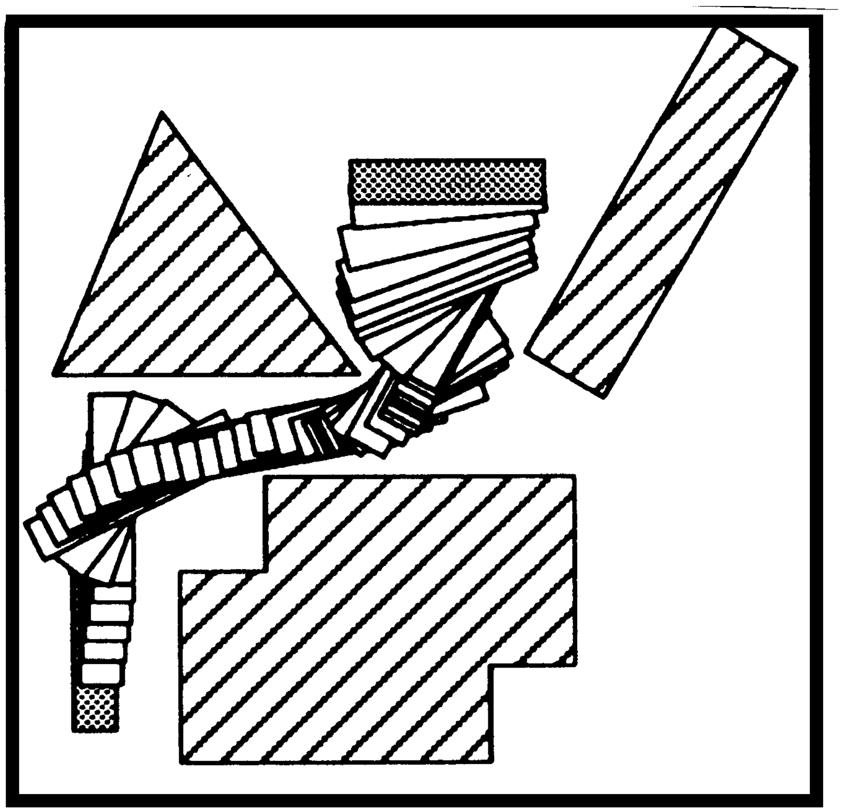
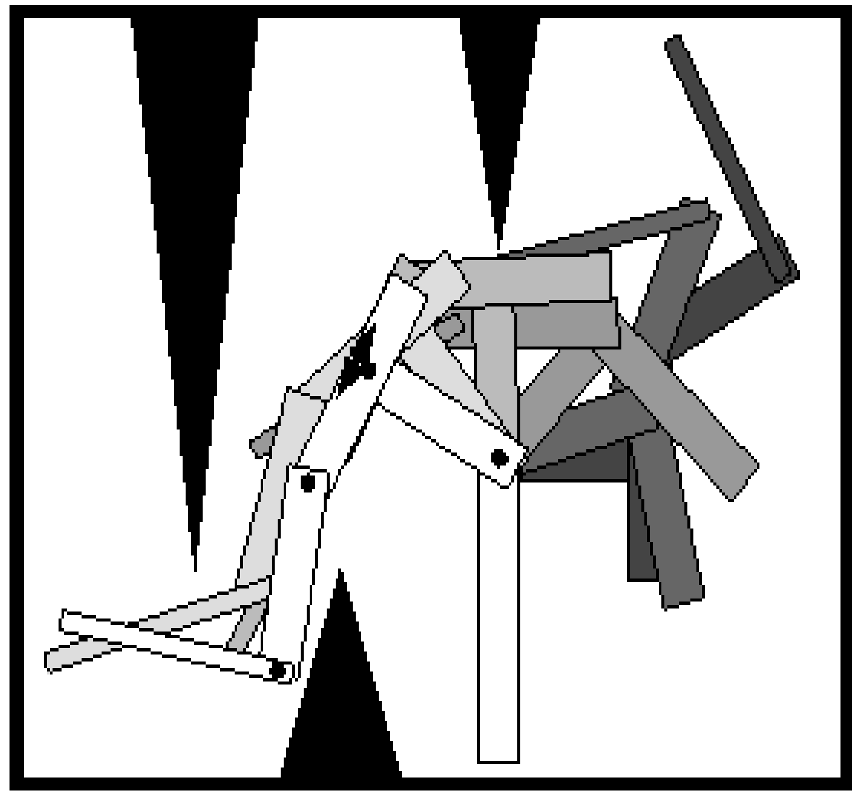


Fig. 3. An example of Voronoi partition using three robots. The working areas of all the robots are represented by different colors. Note that the Voronoi partition is dynamically changing based on different locations of the robots at different time instants, which is a more robust method for dealing with unexpected disturbances in the environments.

What is a Path?



A box robot

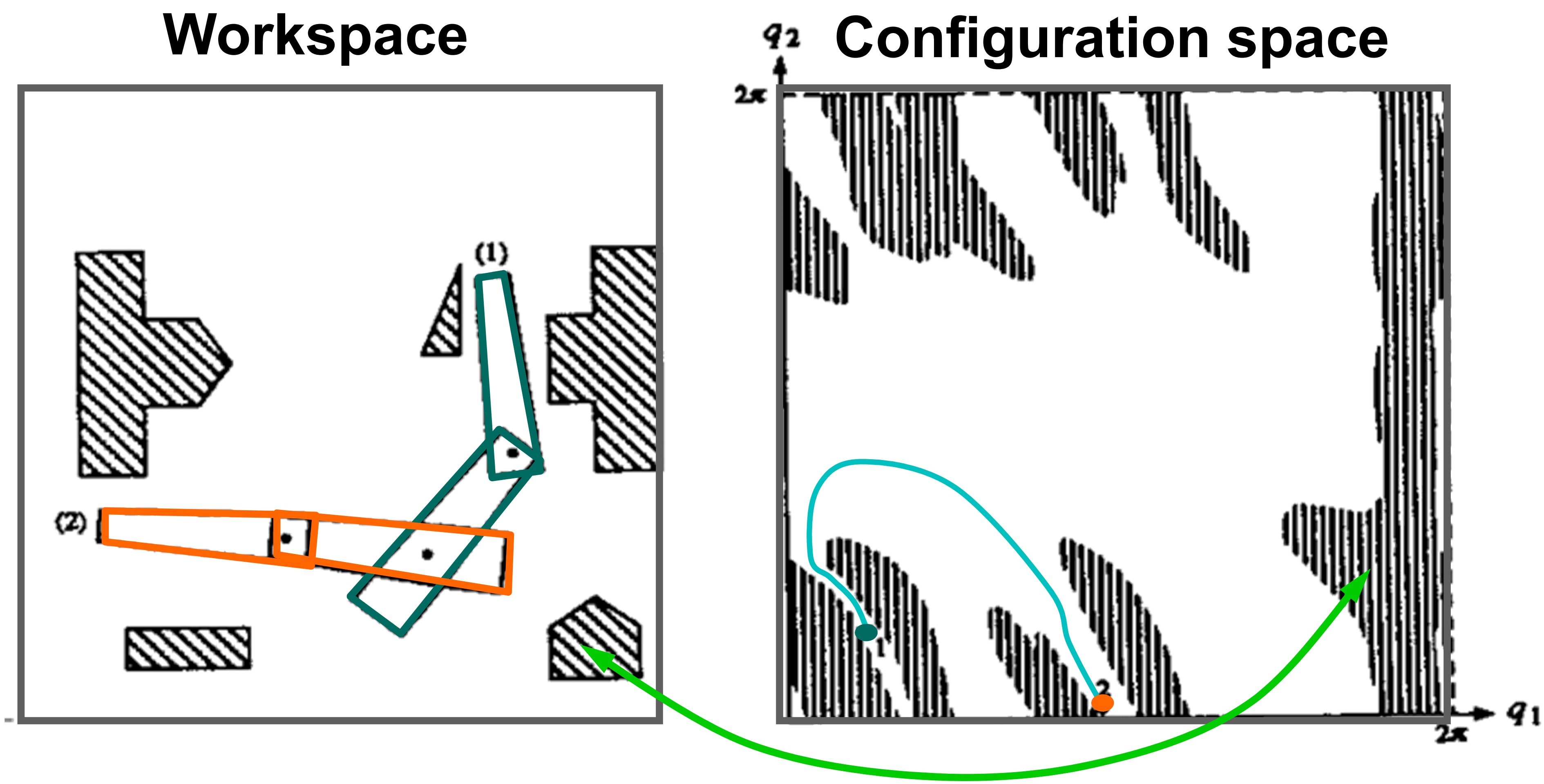


Linked robot

Rough Idea of C-Space

- **Represent degrees-of-freedom (DoFs) of rigid robots, articulated robots, *etc.* into points**
- **Apply algorithms in that space, in addition to the workspace**

Mapping from the Workspace to the Configuration Space



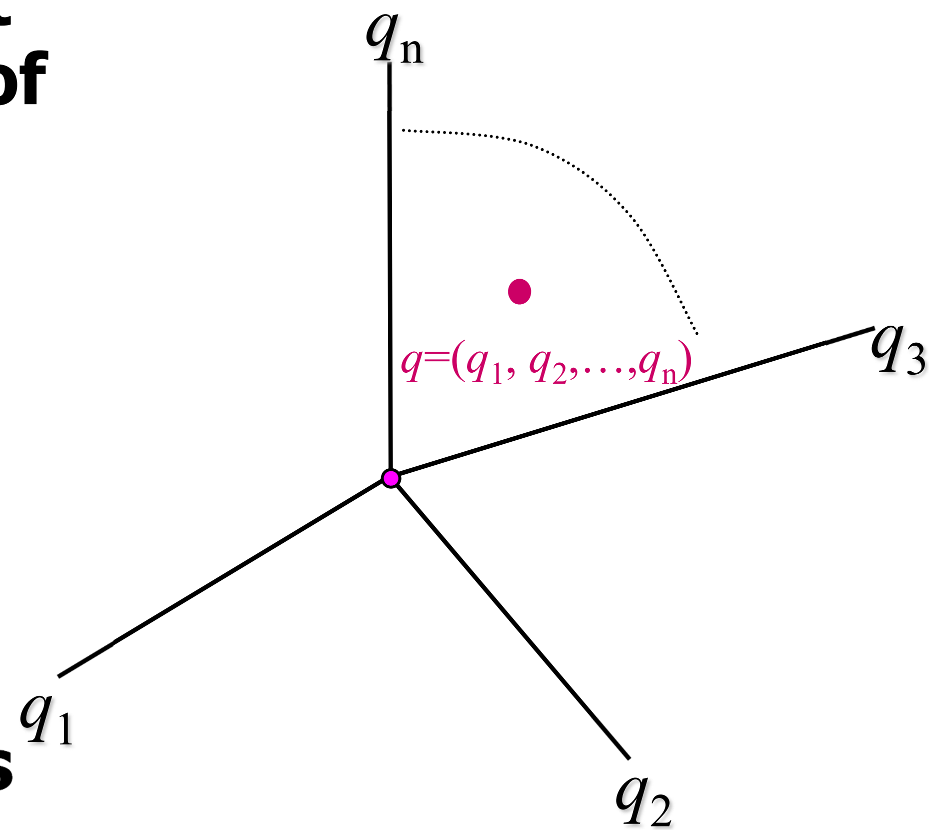
Configuration Space

- Definitions and examples
- Obstacles
- Paths
- Metrics

9

Configuration Space (C-space)

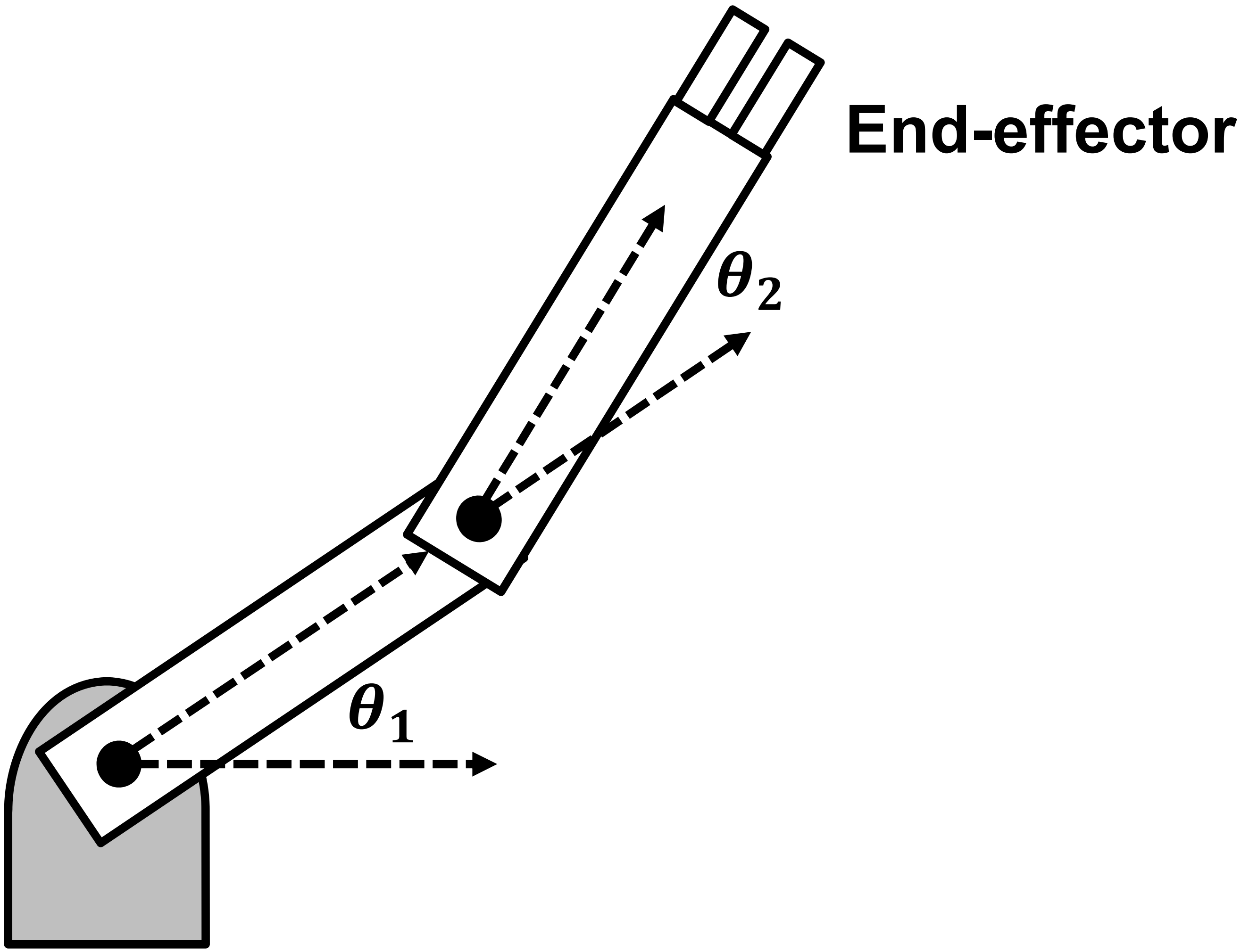
- **The configuration of a robot is a complete specification of the position of every point on the robot**
  - **Usually a configuration is expressed as a vector of position & orientation parameters:** $q = (q_1, q_2, \ldots, q_n)$



- **The configuration space C is the set of all possible configurations**
  - **A configuration is a point in C**

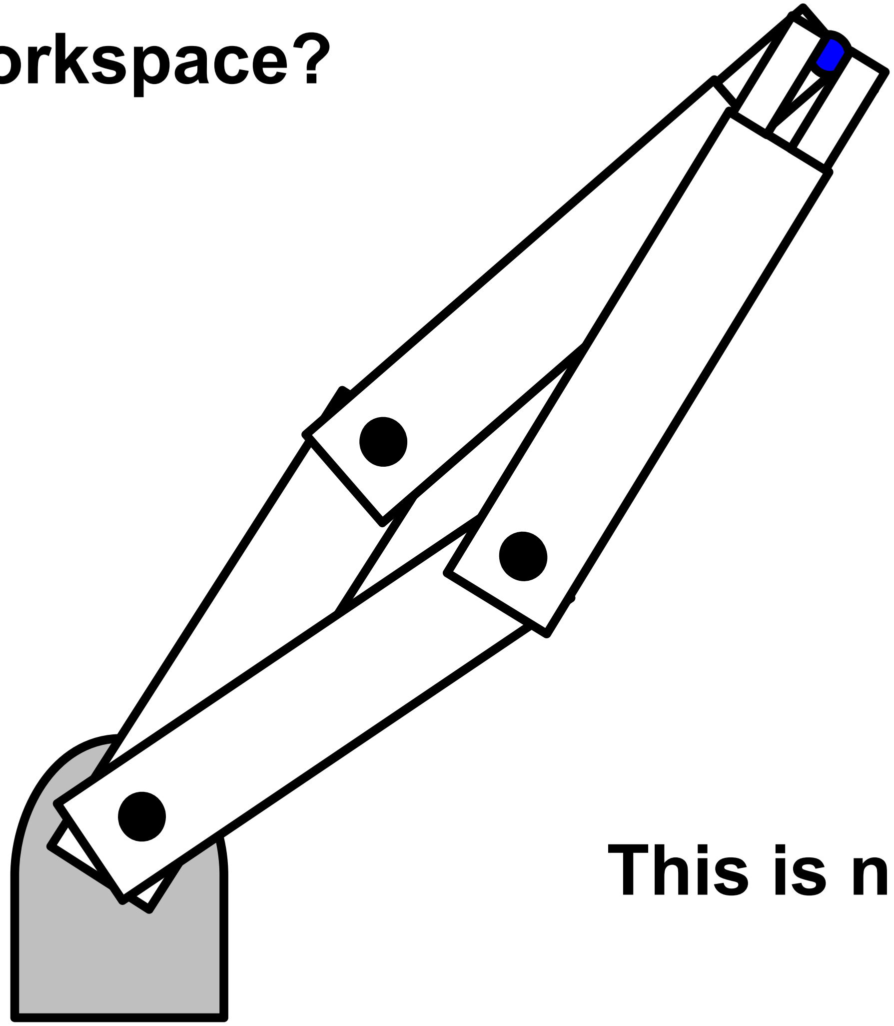
C-space formalism: Lozano-Perez '79

Examples of Configuration Spaces



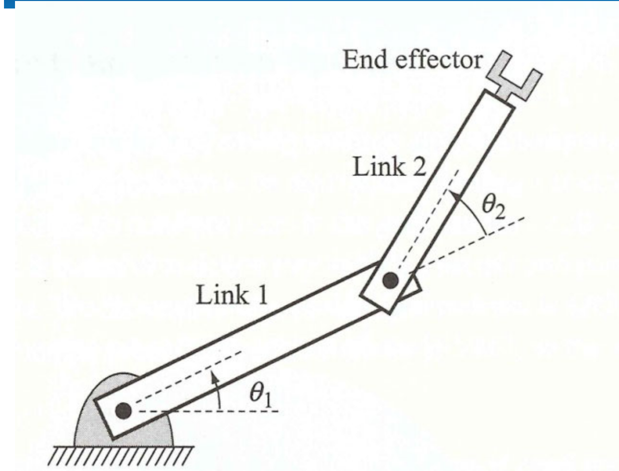
Examples of Configuration Spaces

Consider the end-effector in the workspace?

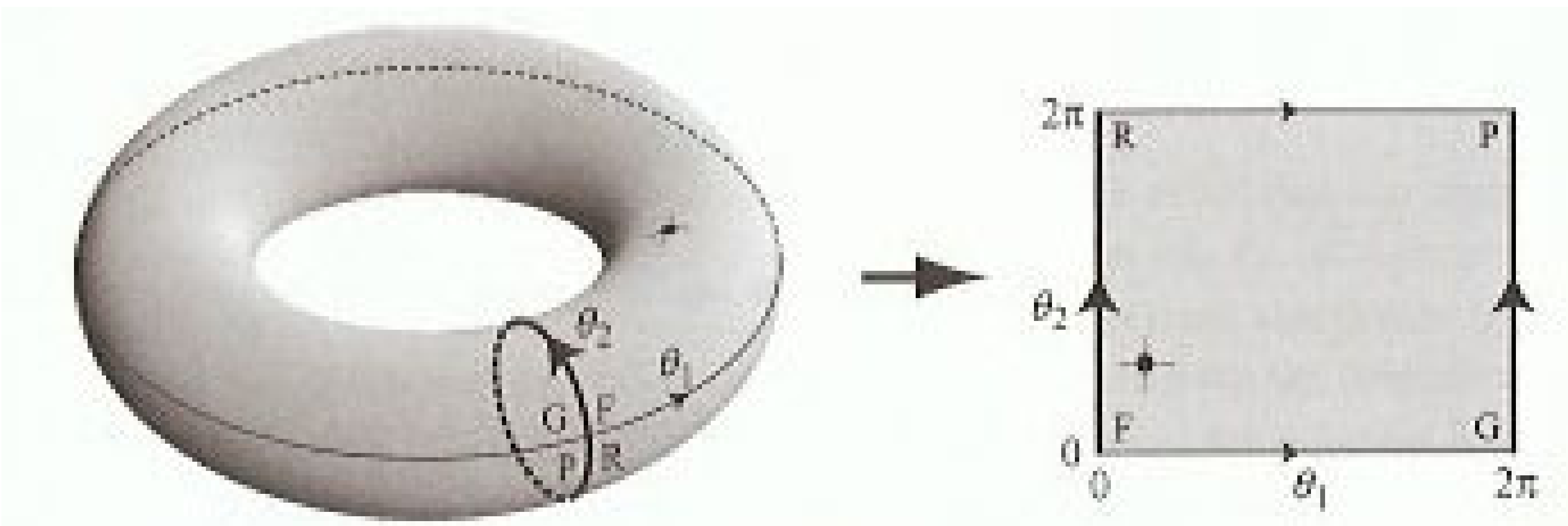


This is not a valid C-space!

Examples of Configuration Spaces

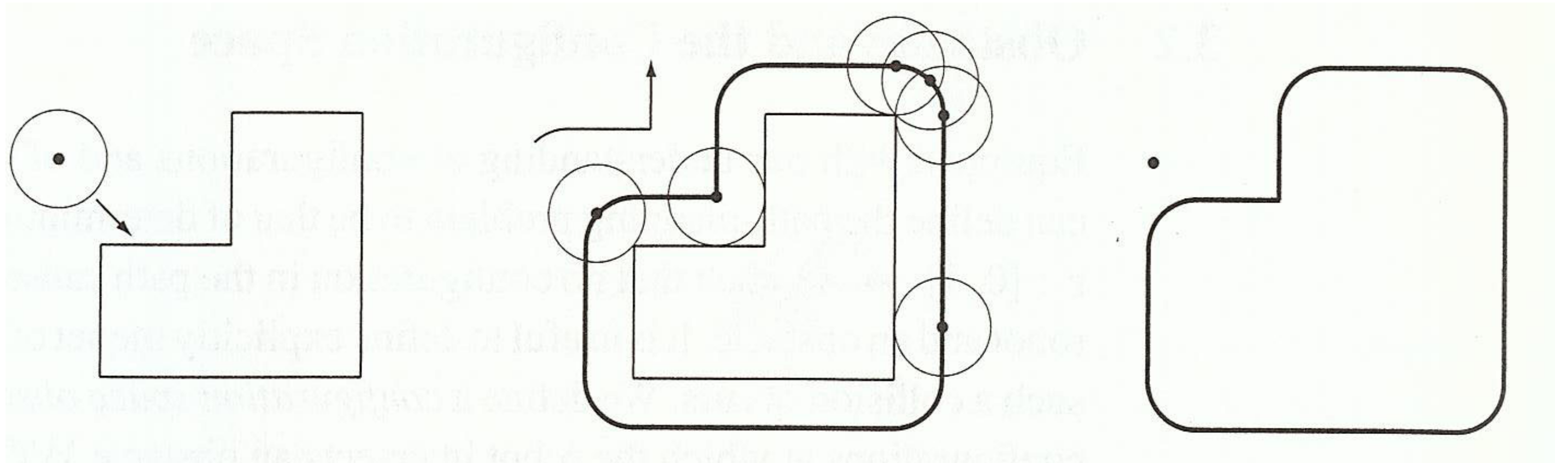


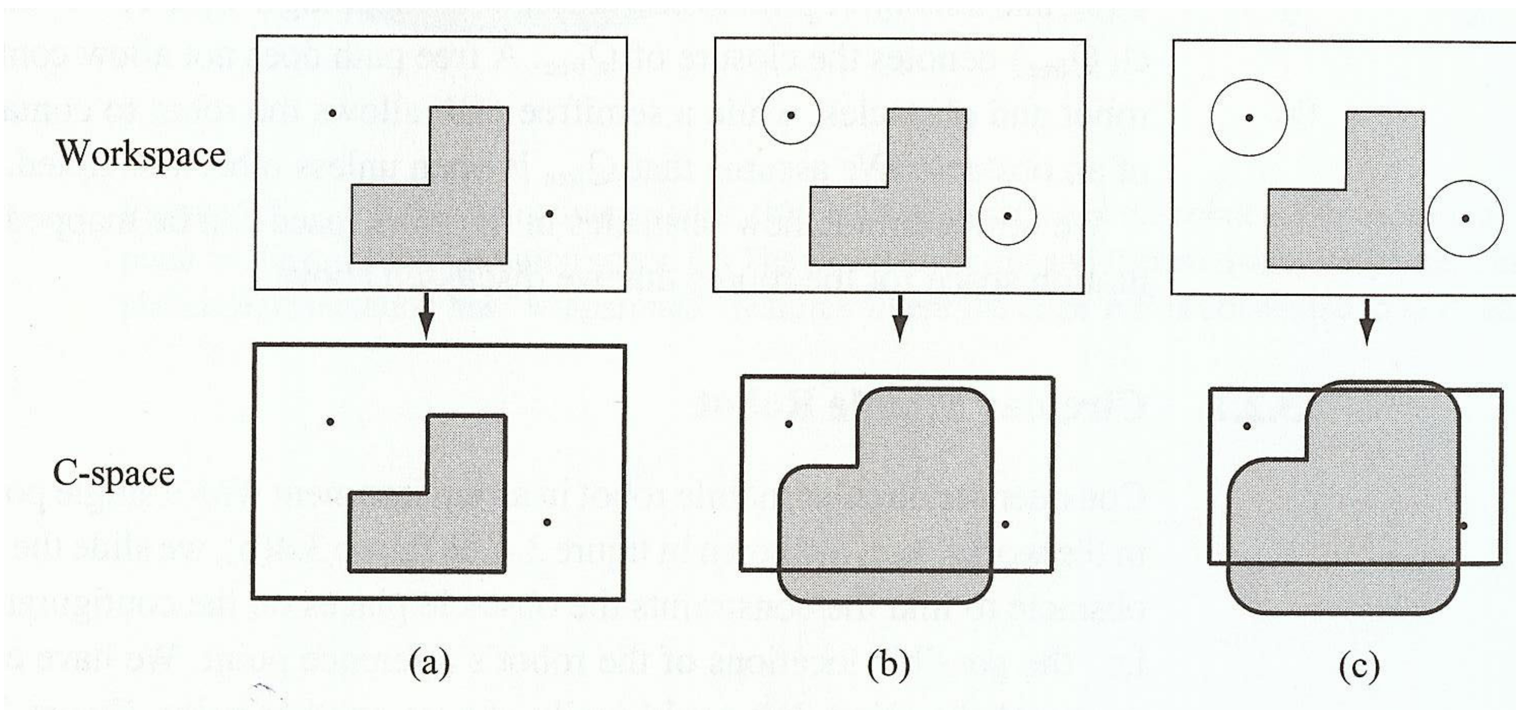
The topology of C is usually **not** that of a Cartesian space R^n.



$S^1 \times S^1 = T^2$

Examples of Circular Robot





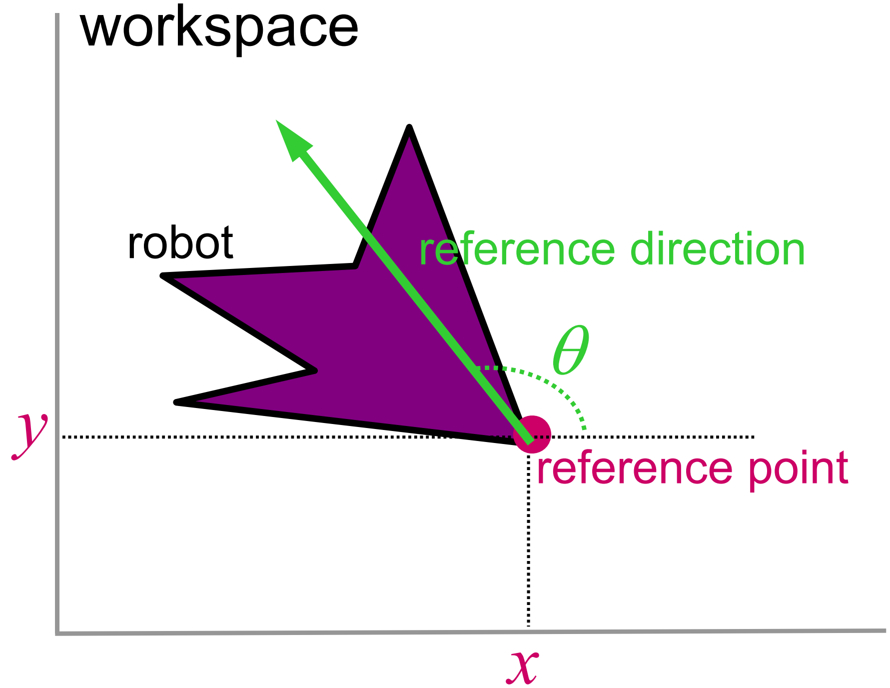
Dimension of Configuration Space

- **The dimension of the configuration space is the minimum number of parameters needed to specify the configuration of the object completely**

- **It is also called the number of degrees of freedom (dofs) of a moving object**

KAIST

Example: Rigid Robot in 2-D Workspace



- **3-parameter specification:** $q = (x, y, \theta)$ with $\theta \in [0, 2\pi)$.
  - **3-D configuration space**

Example: Rigid Robot in 2-D workspace

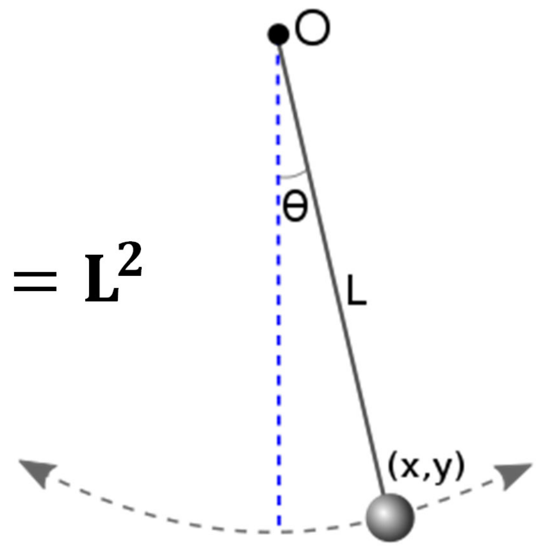
- **4-parameter specification:** $q = (x, y, u, v)$ **with** $u^2+v^2 = 1$. **Note** $u = \cos\theta$ **and** $v = \sin\theta$

- **dim of configuration space = 3**
  - **Does the dimension of the configuration space (number of dofs) depend on the parametrization?**

KAIST

Holonomic and Non-Holonomic Constraints

- Holonomic constraints
  - g (q, t) = 0
  - E.g., pendulum motion: $x^2 + y^2 = L^2$

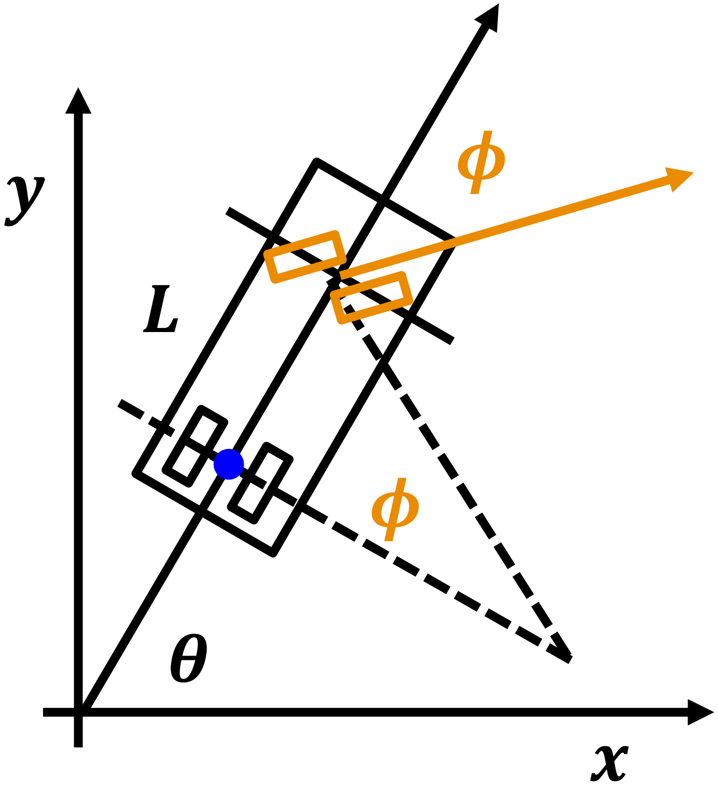


wiki

- Non-holonomic constraints
  - g (q, q', t) = 0 (or q' = f(q, u), where u is an action parameter)
  - This is related to the kinematics of robots
  - To accommodate this, the C-space is extended to include the position and its velocity

Example of Non-Holonomic Constraints

See Kinematic Car Model of my draft



$$\tan(\theta) = \frac{\sin(\theta)}{\cos(\theta)} = \frac{dy}{dx}$$

$$\sin(\theta)dx - \cos(\theta)dy = 0$$

$$\frac{dx}{dt} = v \cdot \cos(\theta), \quad \frac{dy}{dt} = v \cdot \sin(\theta),$$

$$\frac{d\theta}{dt} = \frac{v}{L}\tan(\phi)$$

Note that v, ϕ are action parameters

Holonomic and Non-Holonomic Constraints

- **Dynamic constraints**
  - **Dynamic equations are represented as G(q, q', q'') = 0**
  - **These constraints are reduced to non-holonomic ones when we use the extended C-space such as the state space:**

    S= (X, X'), where X=(q, q')

KAIST

Computation of Dimension of C-Space

- **Suppose that we have a rigid body that can translate and rotate in 2D workspace**
  - **Start with three points: A, B, C (6D space)**

- **We have the following (holonomic) constraints**
  - **Given A, we know the dist to B: d(A,B) = |A-B|**
  - **Given A and B, we have similar equations: d(A,C) = |A-C|, d(B,C) = |B-C|**

- **Each holonomic constraint reduces one dim.**
  - **Not for non-holonomic constraint**

Example: Rigid Robot in 3-D Workspace

- **We can represent the positions and orientations of such robots with matrices (i.e., SO (3) and SE (3))**

SO (n) and SE (n)

- **Special orthogonal group, SO(n),** of n x n matrices R,

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}$$

that satisfy:

$r_{1i}^2 + r_{2i}^2 + r_{3i}^2 = 1$ for all i,

$r_{1i} r_{1j} + r_{2i} r_{2j} + r_{3i} r_{3j} = 0$ for all $i \neq j$,

$\det(R) = +1$

Refer to the 3D Transformation at the undergraduate computer graphics.
http://sgvr.kaist.ac.kr/~sungeui/render/raster/transformation.pdf

- Given the orientation matrix R of SO (n) and the position vector p, **special Euclidean group, SE (n),** is defined as:

$$\begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}$$

Example: Rigid Robot in 3-D Workspace

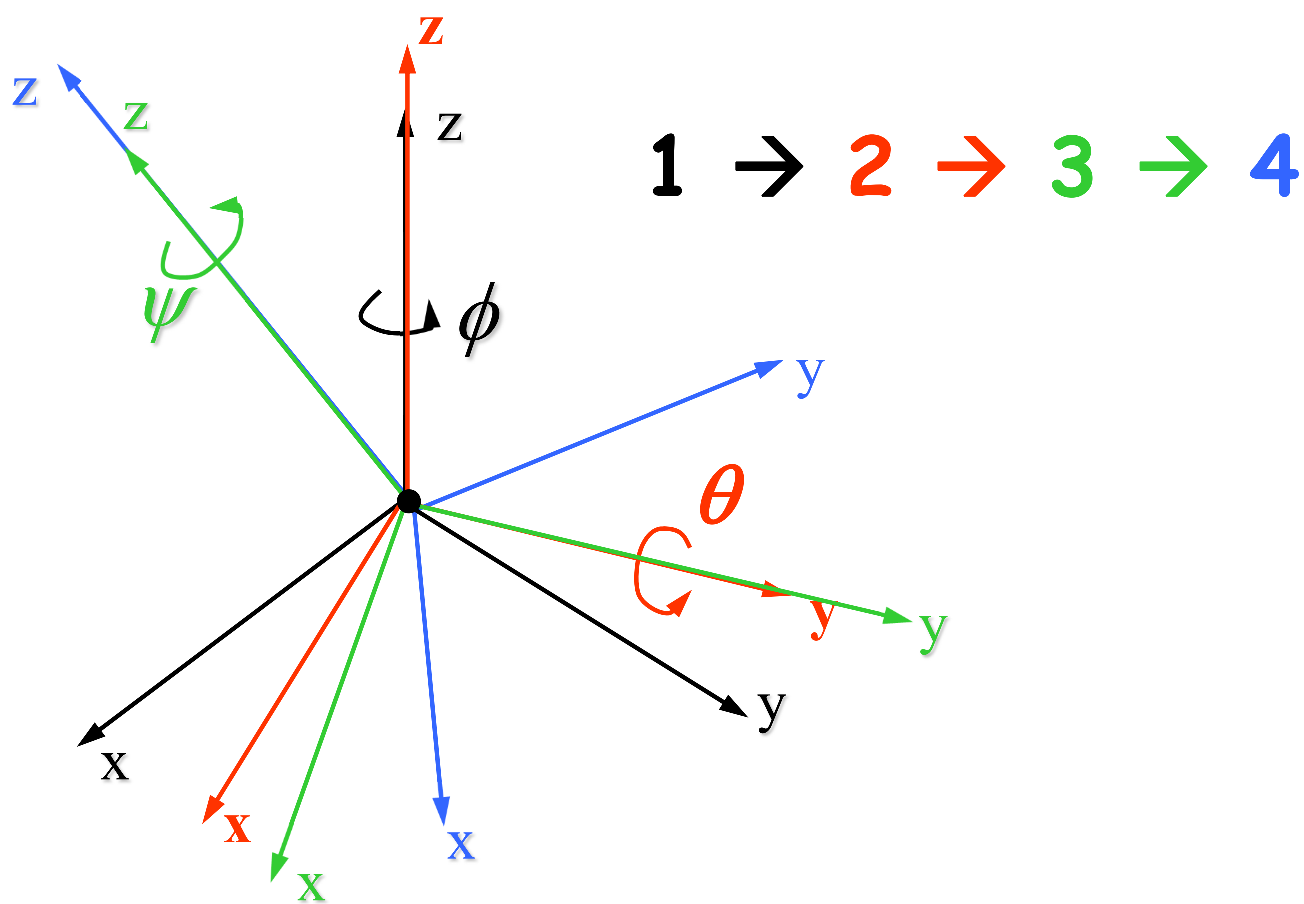
- q = (**position, orientation**) = $(x, y, z, ???)$

- **Parametrization of orientations by matrix:** $q = (r_{11}, r_{12}, \ldots, r_{33}, r_{33})$ **where** $r_{11}, r_{12}, \ldots, r_{33}$ **are the elements of rotation matrix**

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \in SO(3)$$

Example: Rigid Robot in 3-D Workspace

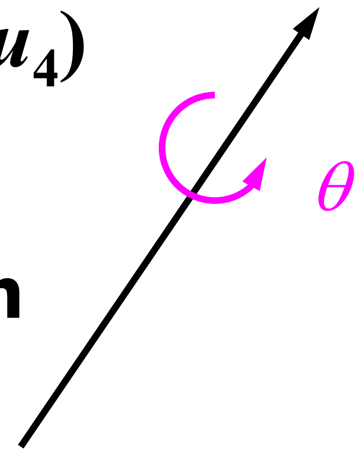
- **Parametrization of orientations by Euler angles:** (ϕ, θ, ψ)



Example: Rigid Robot in 3-D Workspace

- **Parametrization of orientations by unit quaternion:** $u = (u_1, u_2, u_3, u_4)$ **with** $u_1^2 + u_2^2 + u_3^2 + u_4^2 = 1.$
  - **Note** $(u_1, u_2, u_3, u_4) = (\cos\theta/2, n_x\sin\theta/2, n_y\sin\theta/2, n_z\sin\theta/2)$ **with** $n_x^2 + n_y^2 + n_z^2 = 1$

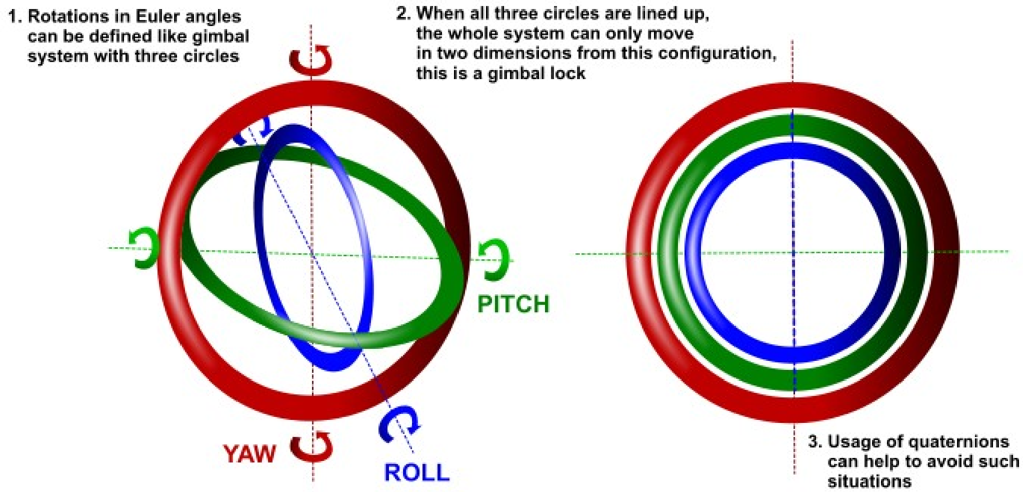
$\mathbf{n} = (n_x, n_y, n_z)$



  - **Compare with representation of orientation in 2-D:** $(u_1, u_2) = (\cos\theta, \sin\theta)$

Example: Rigid Robot in 3-D Workspace

- **Advantage of unit quaternion representation**
  - **Compact**
  - **No singularity (no gimbal lock indicating two axes are aligned)**
  - **Naturally reflect the topology of the space of orientations**
- **Number of dofs = 6**
- **Topology: R^3 x $SO(3)$**



Cyrille Fauvel

Class Objectives were:

- Configuration space
  - Definitions and examples
  - Obstacles
  - Paths
  - Metrics

Next Time….

- Configuration space
  - Definitions and examples
  - Obstacles
  - Paths
  - Metrics

KAIST

Homework

- **Come up with one question on what we have discussed today**
  - **Write a question two times before the mid-term exam**
- **Browse two papers**
  - **Submit their summaries online before the Mon. Class**