CS686: Proximity Queries

Sung-Eui Yoon
(윤성의)

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

KAIST

Presentation Guideline: Expectations

- **Good summary, not full detail, of the paper**
  - **Just 15 min for the talk + quiz**
  - **Talk about motivations of the work**
  - **Give a broad background on the related work**
  - **Explain main idea and results of the paper**
  - **Discuss strengths and weaknesses of the method**

High-Level Ideas

- **Deliver most important ideas and results**
  - **Do not talk about minor details**
  - **Give enough background instead**

- **Spend most time to figure out the most important things and prepare good slides for them**
  - **If possible, re-use existing slides/videos with acknowledgement**

Overall Structure

- **Prepare an overview slide**
  - **Talk about most important things and connect them well**

Be Honest

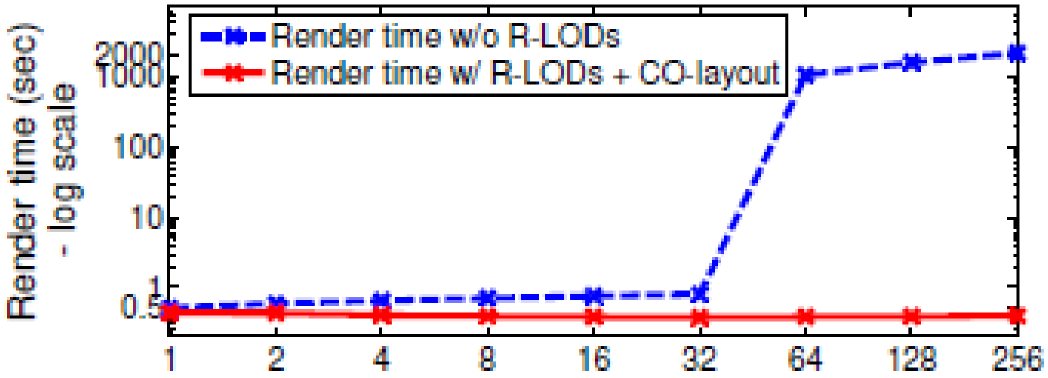
- **Do not skip important ideas that you don't know**
  - **Explain as much as you know and mention that you don't understand some parts**

- **If you get questions you don't know good answers, just say it**
  - **In the end, you need to explain them before the semester ends through KLMS**

KAIST

Result Presentation

- **Give full experiment settings and present data with the related information**



- **After showing the data, give a message that we can pull of the data**
- **Show images/videos, if there are**

Prepare a Quiz

- **Give two simple questions to draw attentions**
  - **Ask a keyword**
  - **Simple true or false questions**
  - **Multiple choice questions**

- **Grade them in the scale of 0 and 10, and send the score to TA**

Audience feedback form

https://forms.gle/tDCDNiJWXNWVQBC56

Date:
Talk title:
Speaker:

A. Was the talk well organized and well prepared?

5: Excellent   4: good   3: okay   2: less than average   1: poor

B. Was the talk comprehensible? How well were important concepts covered?

5: Excellent   4: good   3: okay   2: less than average   1: poor

Any comments to the speaker

KAIST

Science Robotics

Science Robotics | Current Issue | Archive | About | Submit manuscript

SCIENCE ROBOTICS VOLUME 8 | ISSUE 81 | AUG 2023

RESEARCH ARTICLE | BY RACHEL BEATTY, KEEGAN L. MENDEZ, ET AL.

Soft robot–mediated autonomous adaptation to fibrotic capsule formation for improved drug delivery

RESEARCH ARTICLE | BY DAVID A. HAGGERTY, MICHAEL J. BANKS, ET AL.

Control of soft robots with inertial dynamics

RESEARCH ARTICLE | BY DANIEL BRUDER, MORITZ A. GRAULE, ET AL.

Increasing the payload capacity of soft robot arms by localized stiffening

FOCUS | BY TEJAL DESAI, ALESSANDRO GRATTONI

Robotic self-modulation enhances implantable long-acting drug delivery devices

VIEW TABLE OF CONTENTS >

Conf/Journal Ranking

Categories > Engineering & Computer Science > **Robotics**

	Publication	Impact Factor	h5-index	h5-median
1.	IEEE International Conference on Robotics and Automation		119	183
2.	IEEE Robotics and Automation Letters	4.321	106	145
3.	Science Robotics	27.541	98	171
4.	IEEE/RSJ International Conference on Intelligent Robots and Systems		78	137
5.	Robotics and Computer-Integrated Manufacturing		77	117
6.	IEEE Transactions on Robotics	5.567	74	114
7.	IEEE/ASME Transactions on Mechatronics		74	105
8.	The International Journal of Robotics Research		69	109
9.	Robotics: Science and Systems		61	117
10.	Soft Robotics		60	91
11.	Robotics and Autonomous Systems		56	80

Class Objectives (Ch. 4)

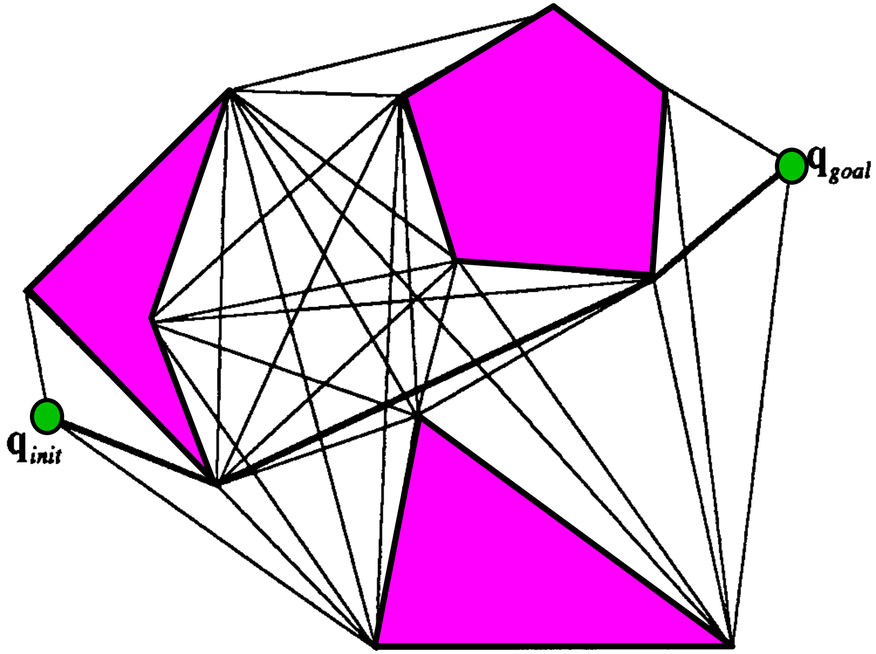
- Understand collision detection and distance computation
  - Bounding volume hierarchies
- Handle point clouds
  - Occupancy map

- Last time:
  - C-obstacle construction using Minkowski sum
  - Homotopy

Two geometric primitives in configuration space

- **CLEAR(*q*)**
  Is configuration *q* collision free or not?

- **LINK(*q*, *q'*)**
  Is the straight-line path between *q* and *q'* collision-free?



Problem

- **Input: two objects A and B**
- **Output:**
  - **Distance computation: compute the distance (in the workspace) between A and B**

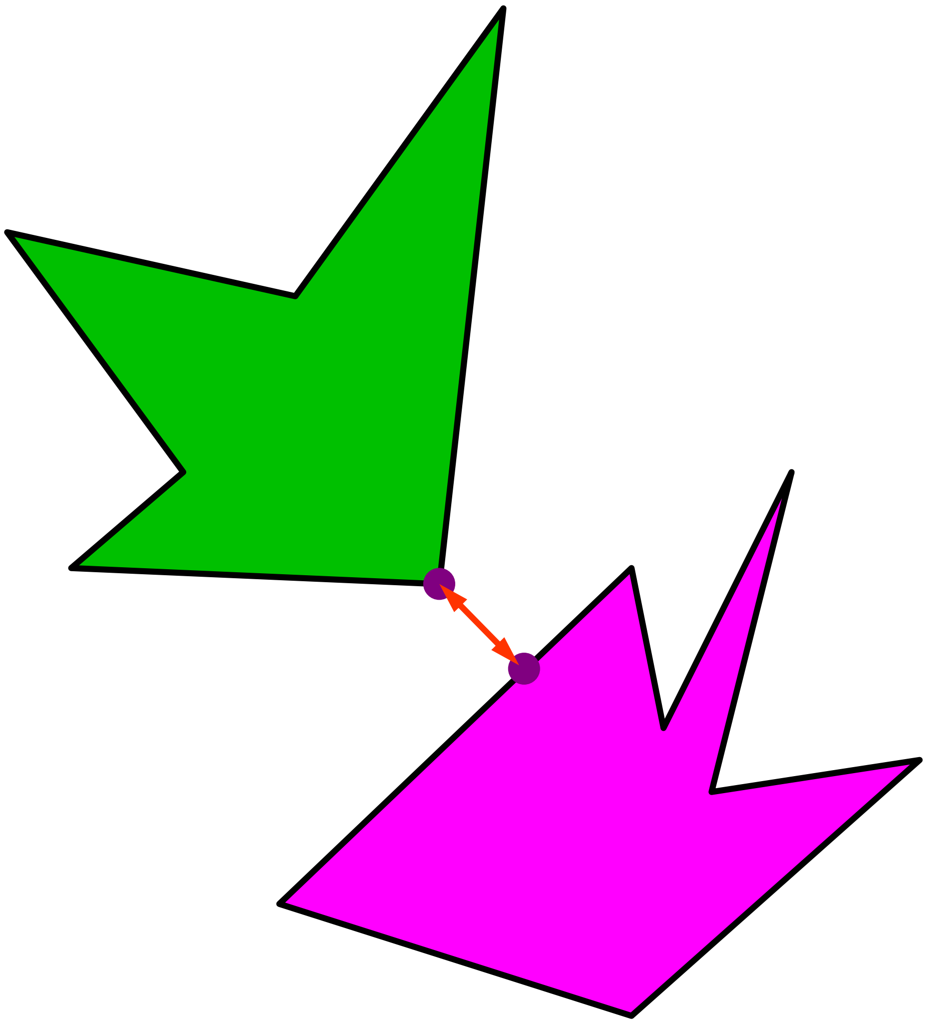
    OR

  - **Collision detection: determine whether A and B collide or not**

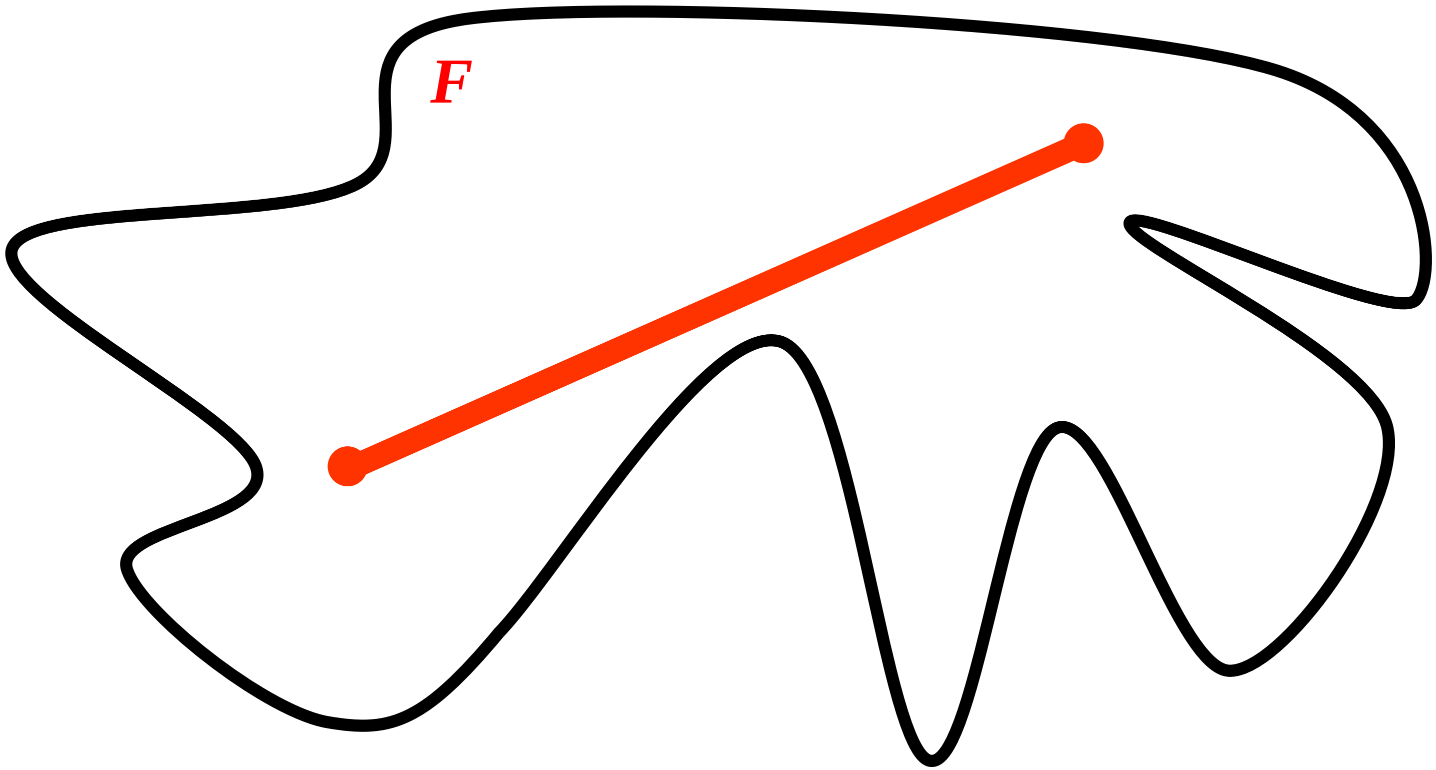
Collision detection vs. distance computation

- **The distance between two objects (in the workspace) is the distance between the two closest points on the respective objects**

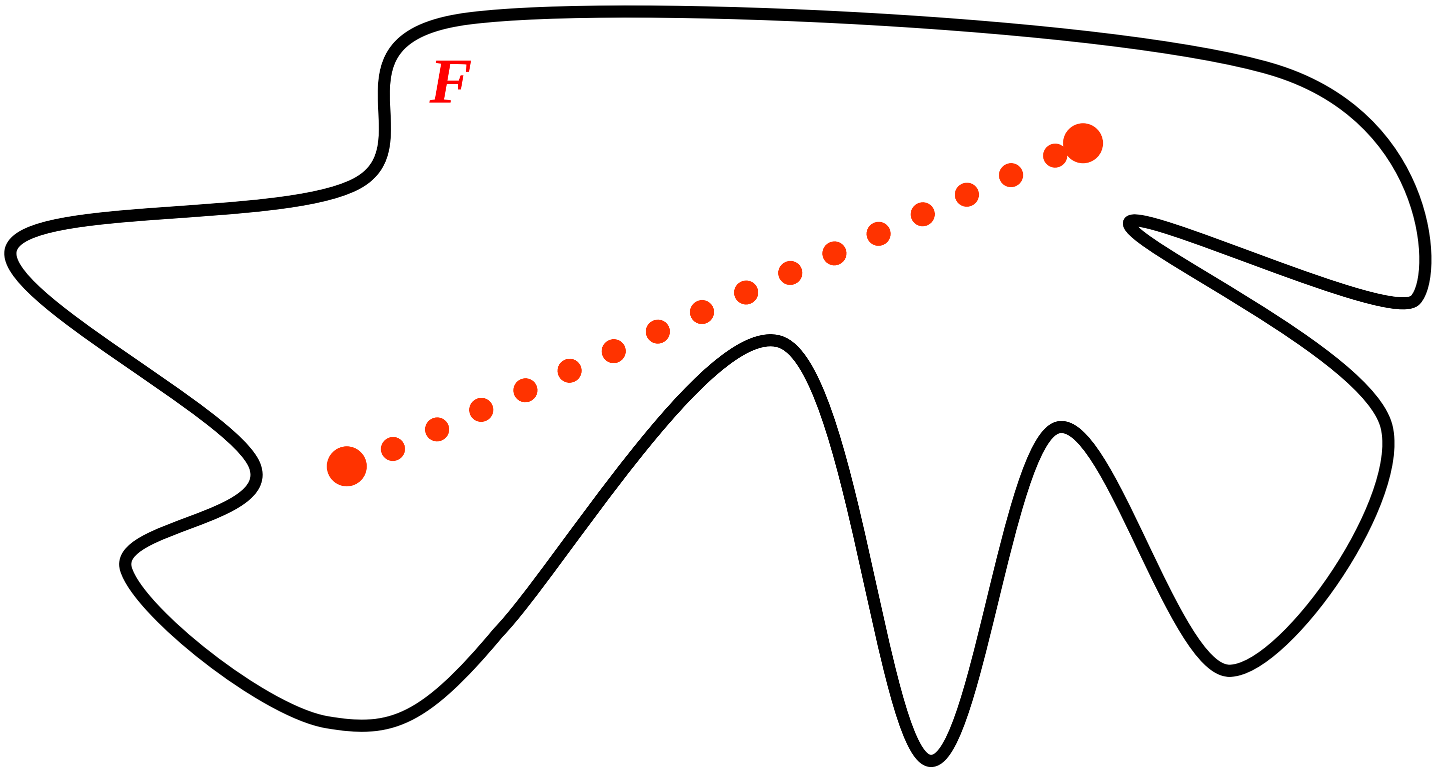
- **Collision if and only if distance = 0**



Collision detection does not allow us to check for free path rigorously

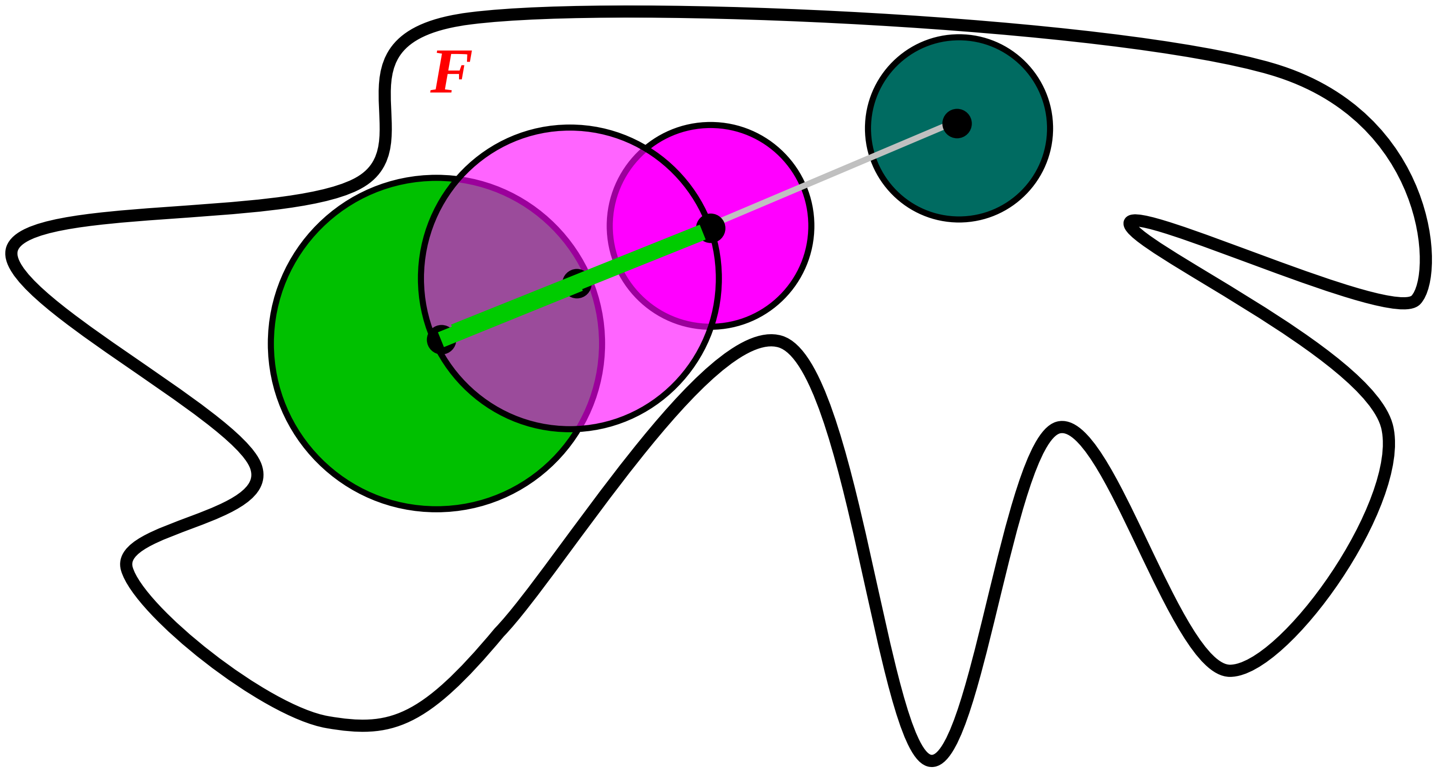


Collision detection does not allow us to check for free path rigorously



Discrete collision checks

Use distance to check for free path rigorously



Use distance to check for free path rigorously

```
Link(q0, q1)
if q0∈N(q1) or q1∈N(q0)
then
    return TRUE.
else
    q′ = (q0+q1)/2.
    if q′ is in collision
    then
        return FALSE
    else
        return Link(q0, q′) && Link(q1,q′).
```

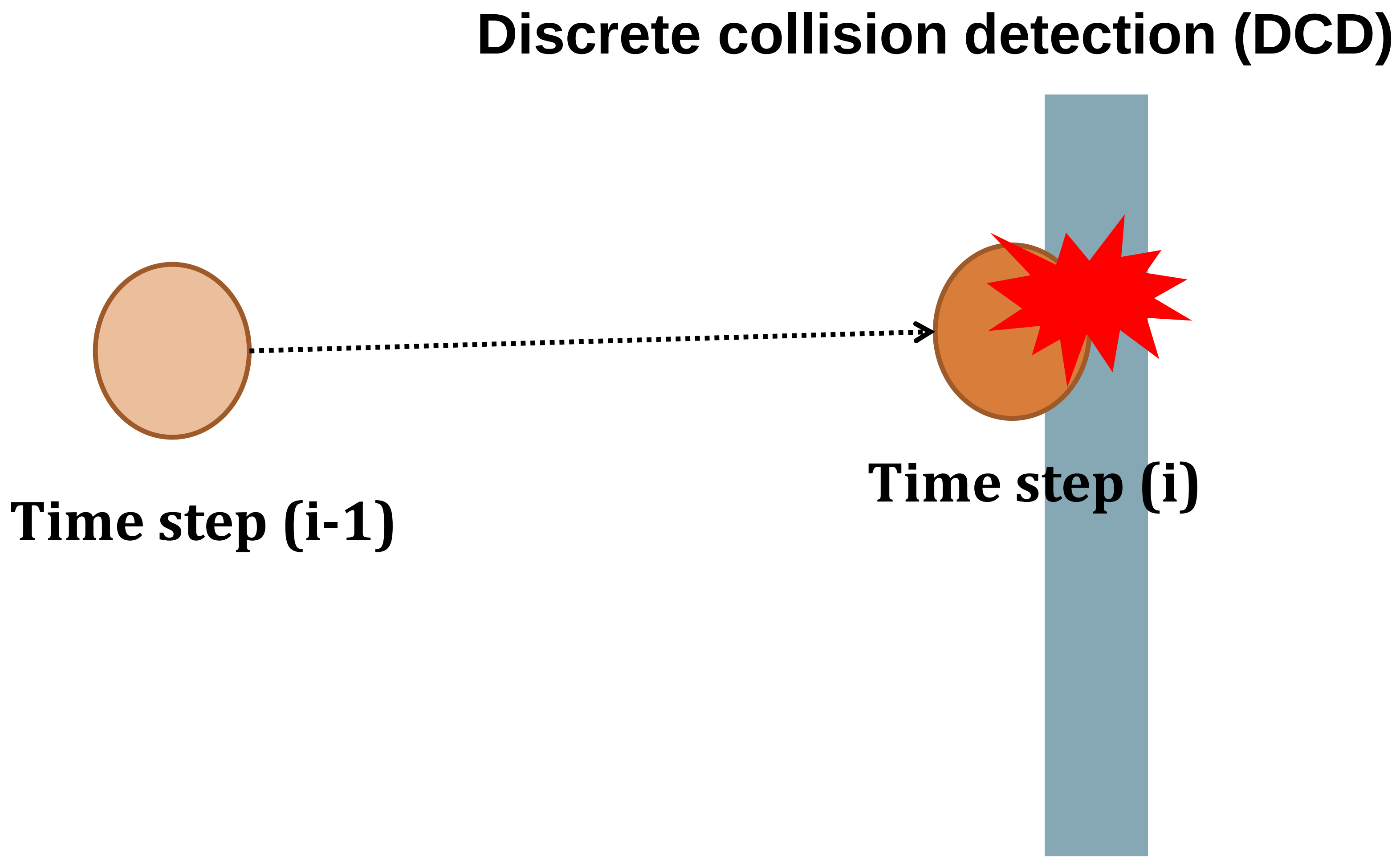
Applications

- Robotics
  - Collision avoidance
  - Path planning
- Graphics & virtual environment simulation
- Haptics
  - Collision detection
  - Force proportional to distance

Collision Detection

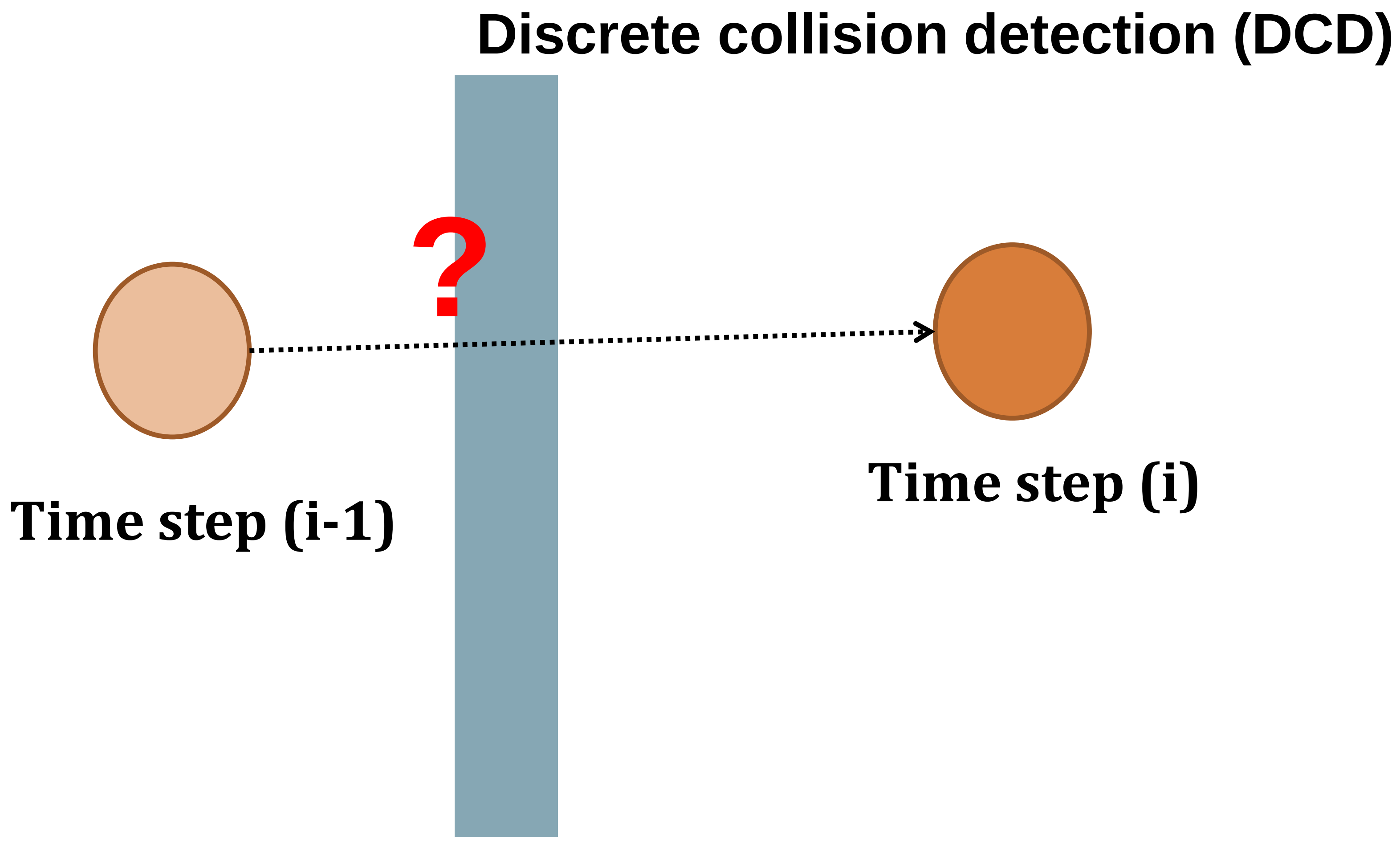
- **Discrete collision detection**
- **Continuous collision detection**

Discrete VS Continuous



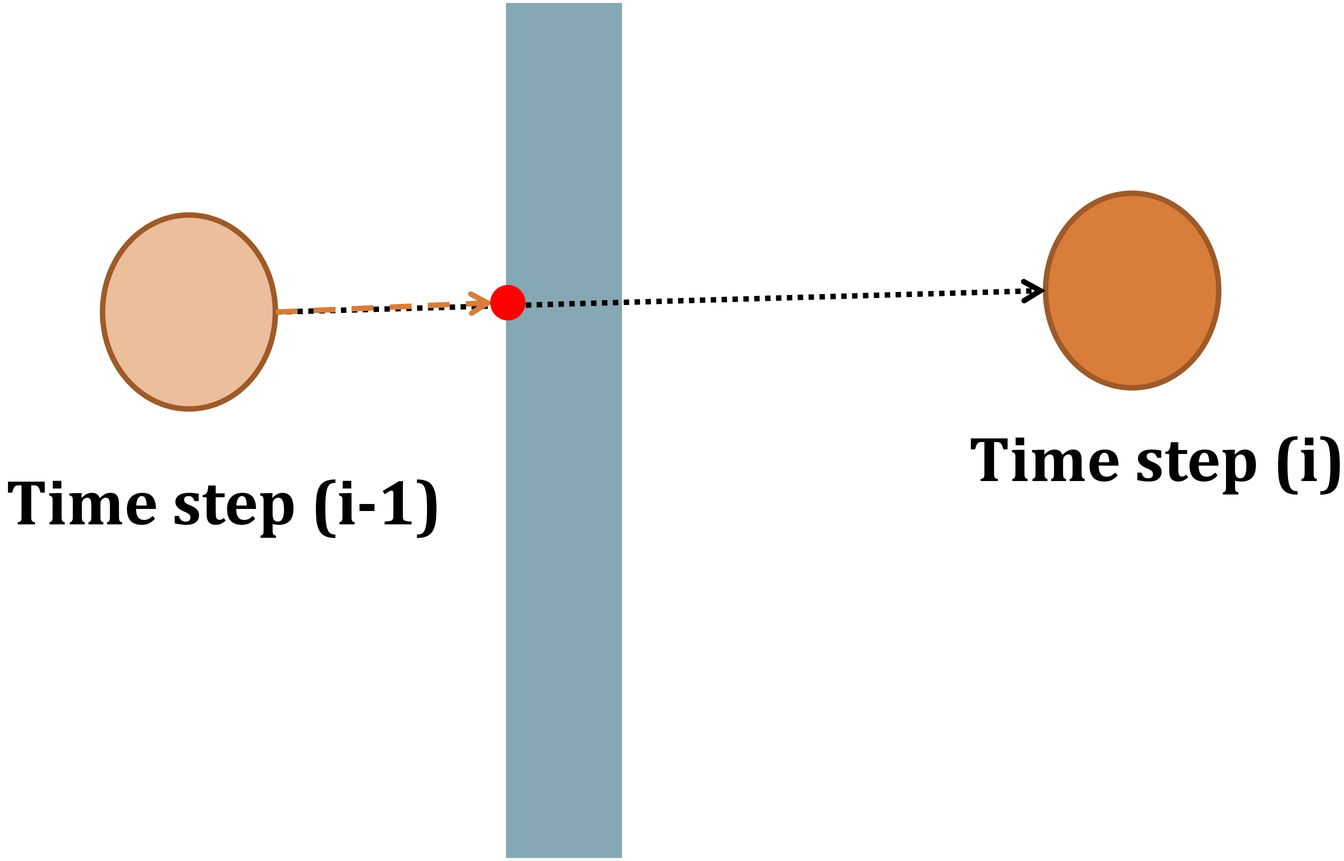
From Duksu's slides

Discrete VS Continuous



Discrete VS Continuous

Continuous collision detection(CCD)



Discrete VS Continuous

	Continuous CD	**Discrete CD**
Accuracy	Accurate	May miss some collisions
Computation time	Slow	Fast

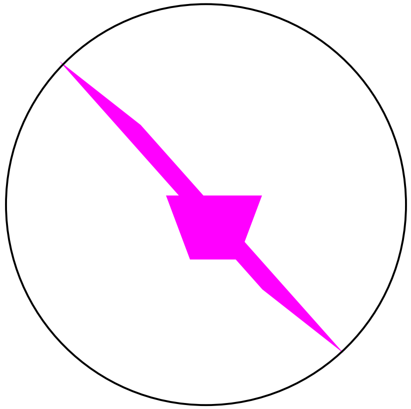
Collision Detection

- **Discrete collision detection**
- **Continuous collision detection**

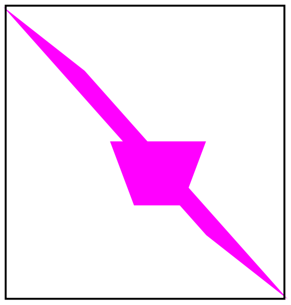
- **These are typically accelerated by bounding volume hierarchices (BVHs)**

Bounding Volumes

- Sphere [Whitted80]
  - Cheap to compute
  - Cheap test
  - Potentially very bad fit

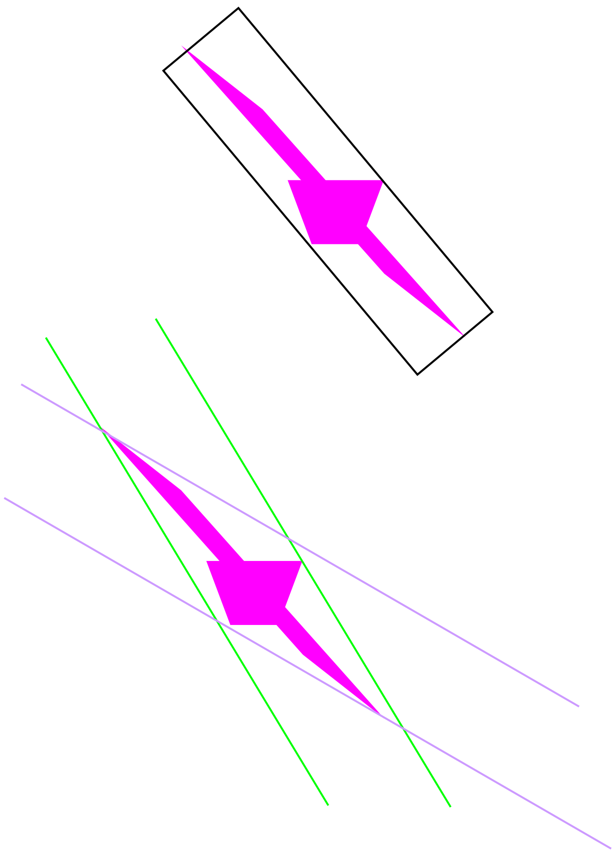


- Axis-aligned bounding box
  - Very cheap to compute
  - Cheap test
  - Tighter than sphere



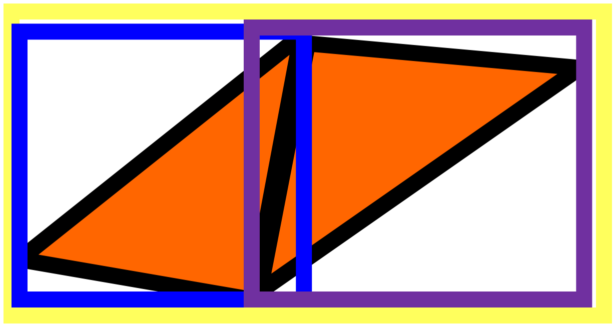
Bounding Volumes

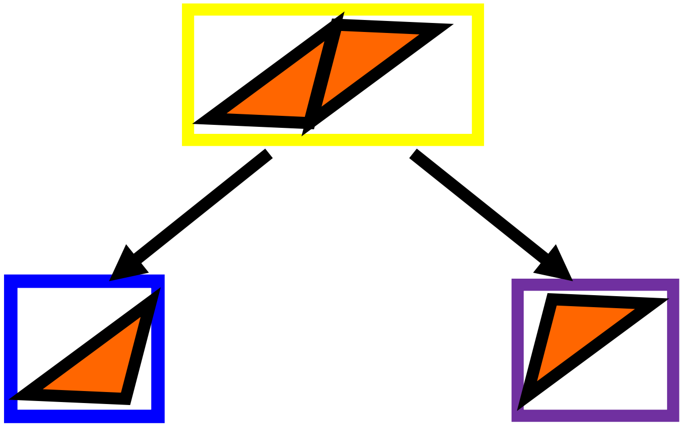
- **Oriented bounding box**
  - **Fairly cheap to compute**
  - **Fairly cheap test**
  - **Generally fairly tight**
- **Slabs / K-dops**
  - **More expensive to compute**
  - **Fairly cheap test**
  - **Can be tighter than OBB**



Bounding Volume Hierarchies (BVHs)

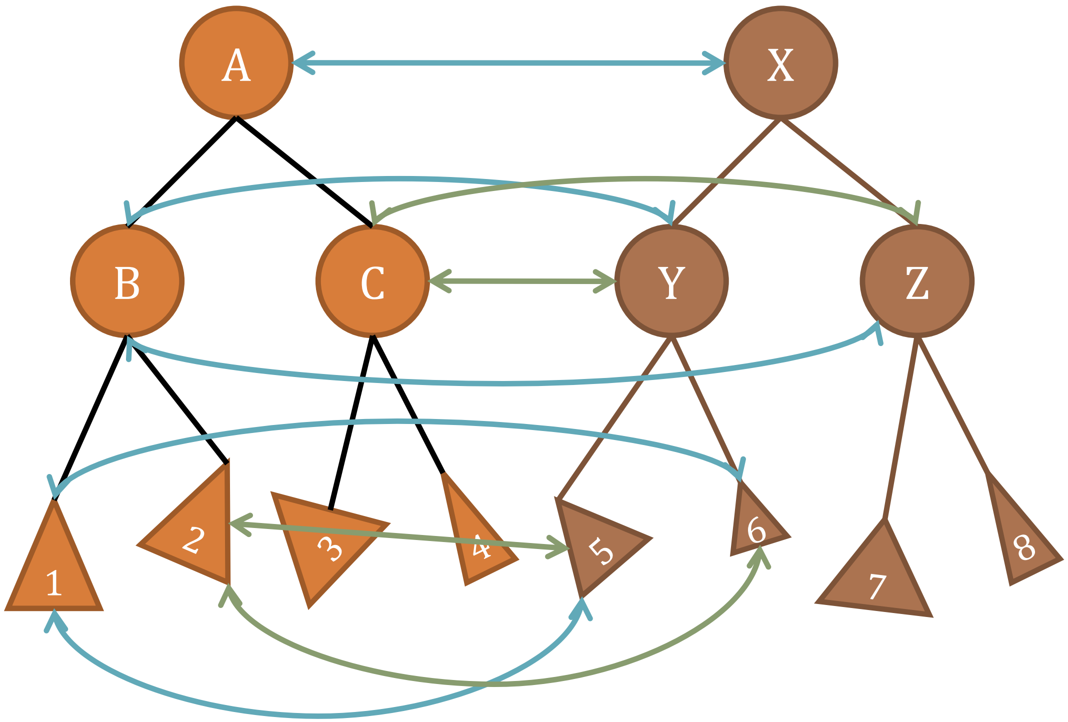
- **Organize bounding volumes recursively as a tree**
- **Construct BVHs in a top-down manner**
  - **Use median-based partitioning or other advanced partitioning methods**





A BVH

Collision Detection with BVHs



BV overlap test (A,X)

(A,X)

BV overlap test
(B,Y), (B,Z), (C,Y), (C,Z)

(B,Y)

Primitive collision test
(1,5), (1,6), (2,5), (2,6)

(1,5)

Triangle 1 and 5 have a collision!

From Duksu's slides

Test-Of-Time 2006 Award

RT-DEFORM: Interactive Ray Tracing of Dynamic Scenes using BVHs

Christian Lauterbach, Sung-eui Yoon, David Tuft, Dinesh Manocha

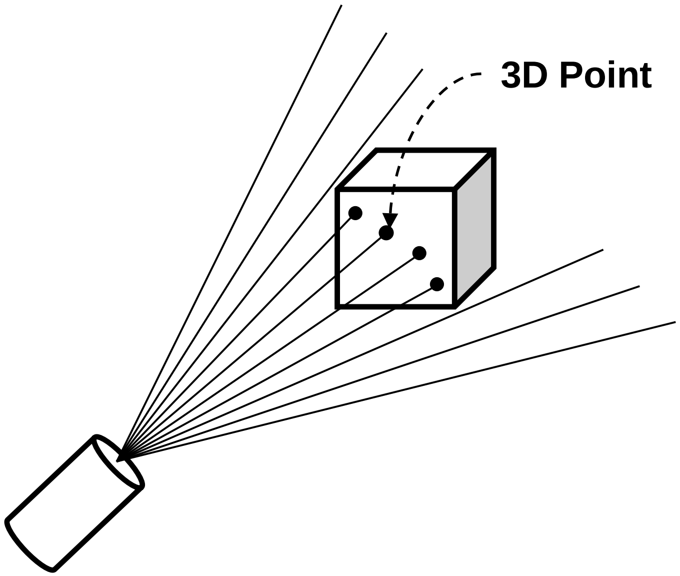
IEEE Interactive Ray Tracing, 2006


RT-DEFORM: Interactive Ray Tracing of Dynamic Scenes using BVHs

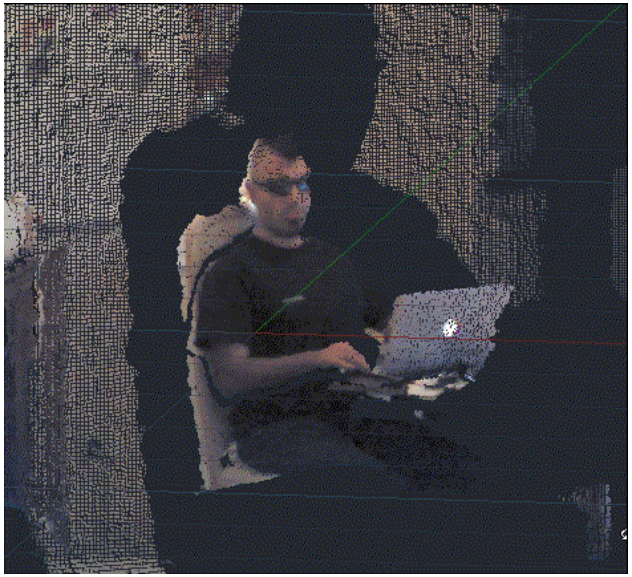
Christian Lauterbach, Sung-Eui Yoon, David Tuft, Dinesh Manocha

3D Sensor & Point Cloud Data

- 3D sensor generates excessive amount of points with some noise periodically
  - 300K points / 30FPS with Kinect



3D Sensor Model



Point Cloud Data

Modified from YongSun's slides

KAIST

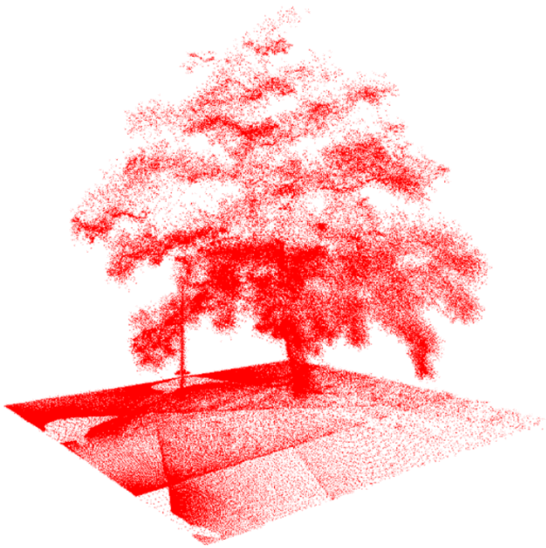
Sensor-based Path Planning

- **Navigation using 3D depth sensor**

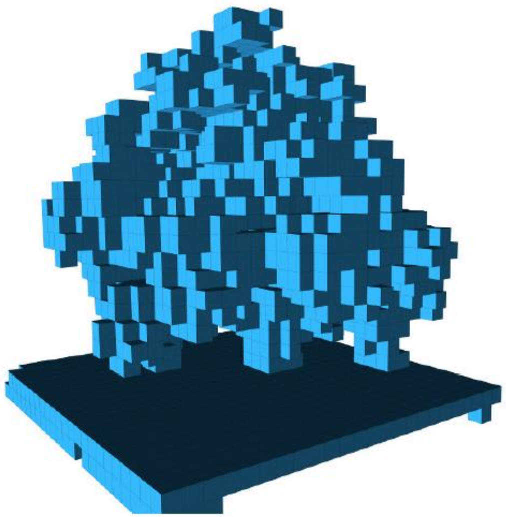
Video

KAIST

General Flow of Using Point Clouds



Update

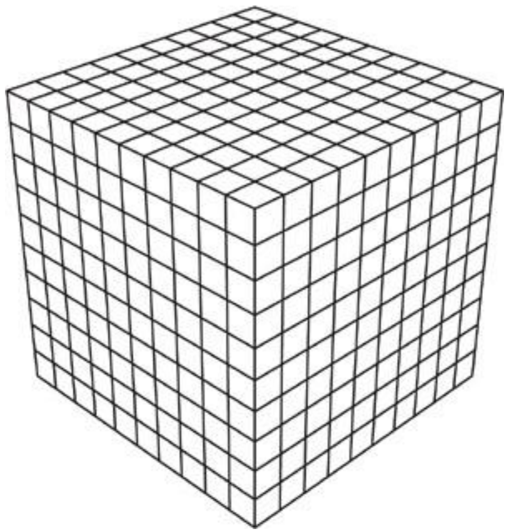


Point cloud

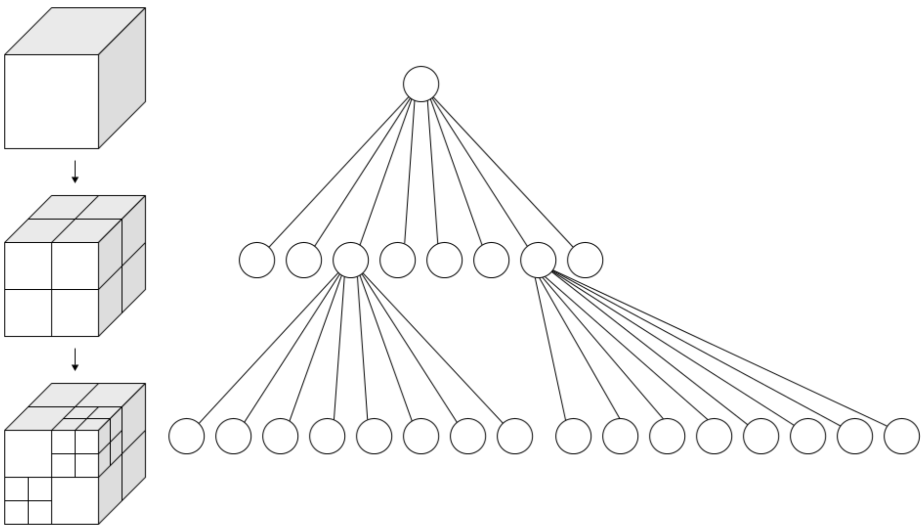
Maps (octree or grid)

Applications (path planning)

Map Representations



3D Grid Map



Octree Data Structure

Occupancy Map Representation

- **OctoMap [Wurm et al., *ICRA, 2010*]**
  - **Encode an occupancy probability of cell n given measurement $z_{1:t}$**

$$\boldsymbol{L(n \mid z_{1:t}) = L(n \mid z_{1:t-1}) + L(n \mid z_t)}$$

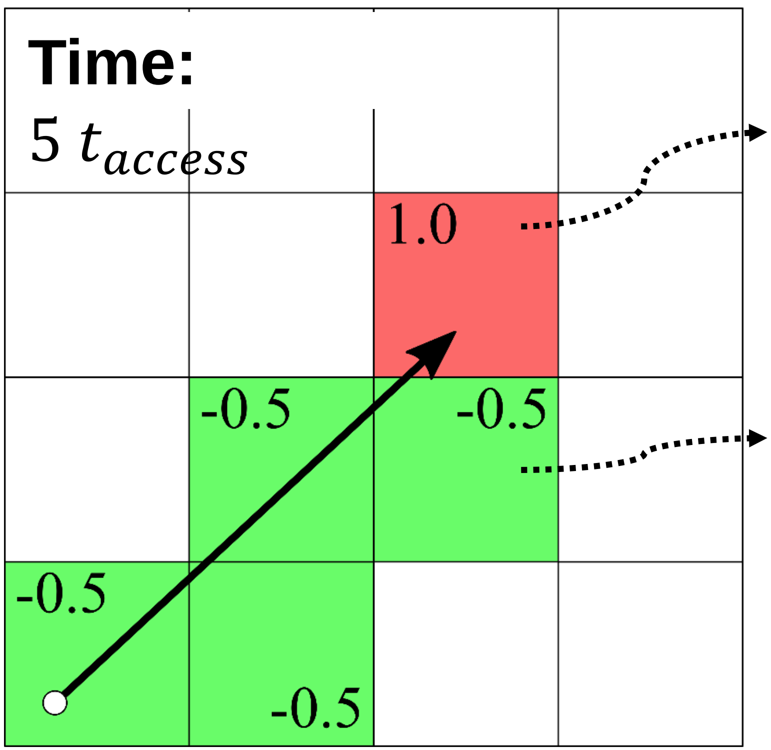
Occupancy probability of the cell n at time step $t-1$

New sensor measurement z_t to be updated at time step t

$$L(n \mid z_t) = \begin{cases} l_{occ} & occupied\ state \\ l_{free} & free\ state \end{cases}$$

Update Method

- **Traverse and update cells**
  - **Bresenham algorithm [Amanatides et al., *Eurographics, 1987*]**



Updated cell to occupied state

$L(n \mid z_t) = l_{occ} = 1.0$

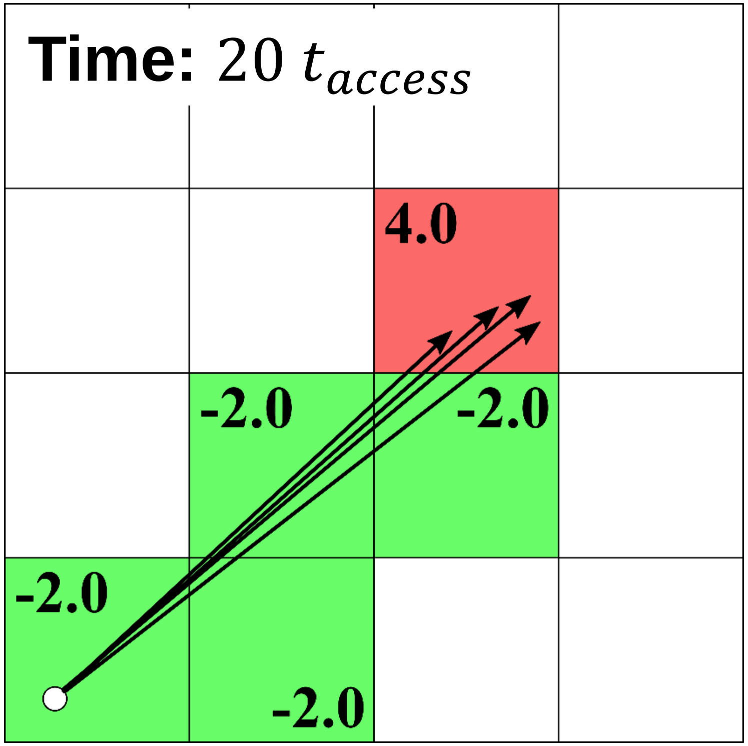
Updated cell to free state

$L(n \mid z_t) = l_{free} = -0.5$

t_{access}**: time to update a cell**

Update Method

- Traverse and update cells
  - Bresenham algorithm [Amanatides et al., *Eurographics, 1987*]
  - Can be very slow, with many points

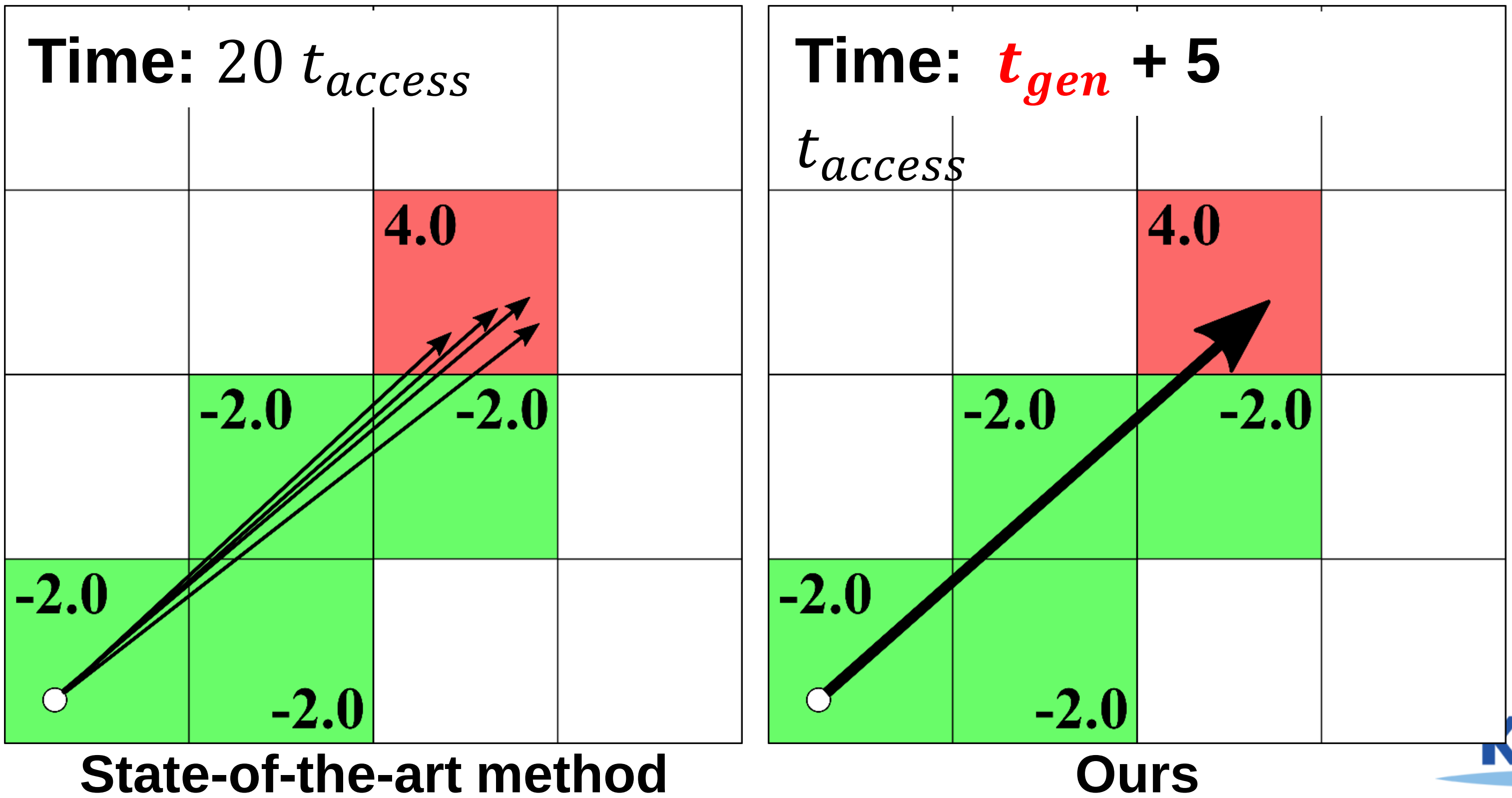


- Visit **the same cells multiple times** for multiple rays

t_{access}: time to update a cell

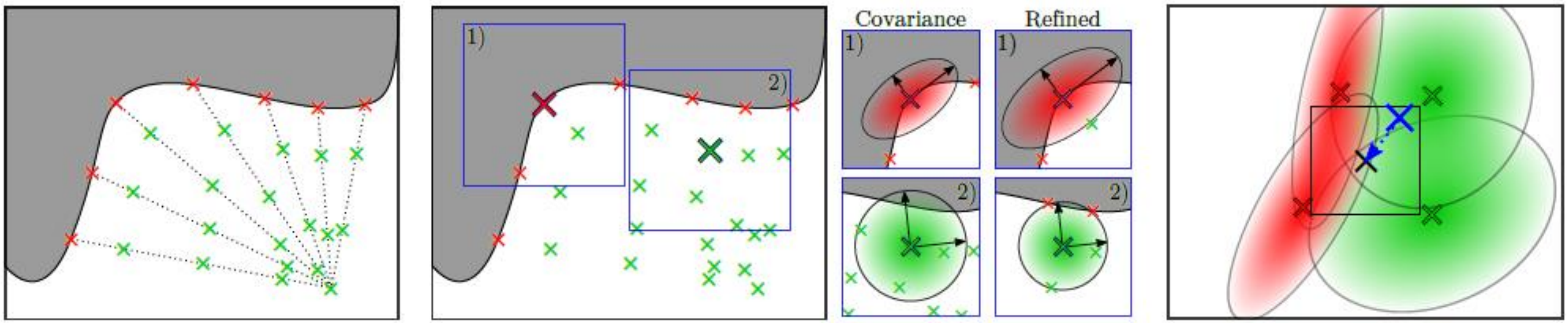
Super Rays [Kwon et al., ICRA16]

- **Benefits of our approach**
  - **Faster performance with the same representation accuracy**
  - **Codes are available**



Learning based Approaches

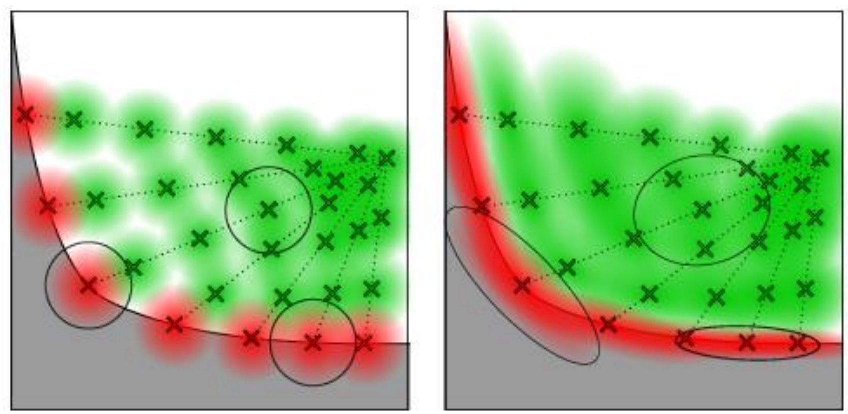
- **Unobserved regions due to occlusion and sensor errors**



(a) Occupancy sampling (b) Adaptive bandwidth selection (c) Kernel estimation update

- **Estimate status of such regions based on learning techniques**
  - **[Kwon et al., IROS 20]**

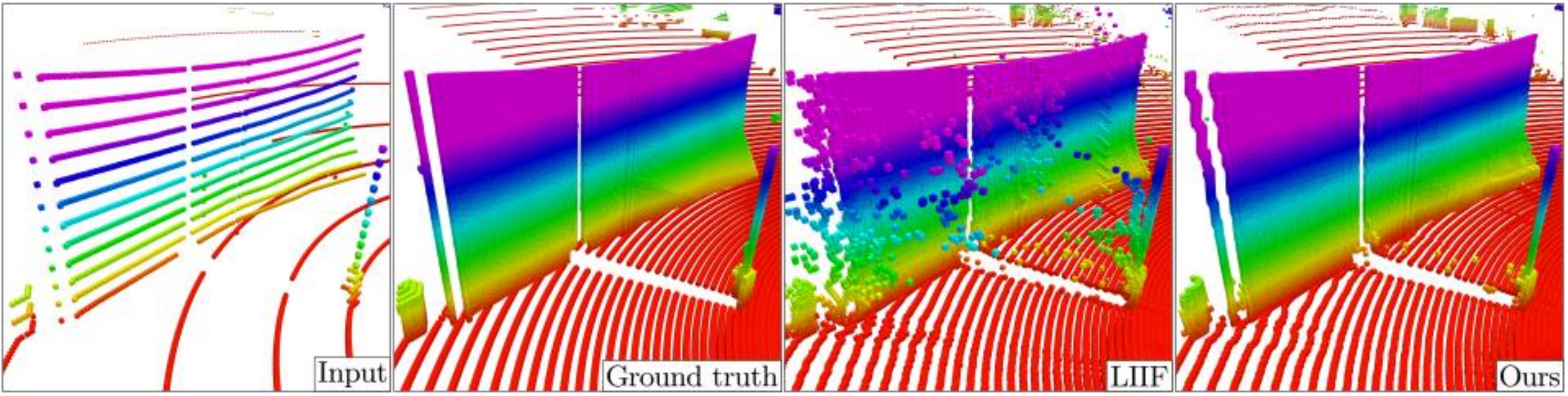
https://sgvr.kaist.ac.kr/~yskwon/papers/iros20-akimap/



(a) Fixed kernel inference (b) Adaptive kernel inference

LiDAR Super-Resolution

- **Implicit LiDAR Network: LiDAR Super-Resolution via Interpolation Weight Prediction, ICRA 22**



https://sgvr.kaist.ac.kr/~yskwon/papers/icra22-iln/

KAIST

Class Objectives were:

- **Understand collision detection and distance computation**
  - **Bounding volume hierarchies**
- **Handle point clouds**
  - **Occupancy map**
- **Ch. 4 of my book**

Next Time…

- **Probabilistic Roadmaps**

Homework

- Submit summaries of 2 ICRA/IROS/RSS/CoRL/TRO/IJRR papers
- Go over the next lecture slides
- Come up with two questions before the mid-term exam