High-Performance Geometric Computation for Robot Motion Planning

김영준 (Young J. Kim)
Computer Science and Engineering
Ewha Womans University

Agenda

- Collision connection query (CCQ)
  - Sampling-based motion planning
  - Forward grasp planning
- Penetration query (PD)
  - Sampling-based motion planning
  - Optimization-based motion planning

Available as FCL package in ROS (http://wiki.ros.org/fcl)

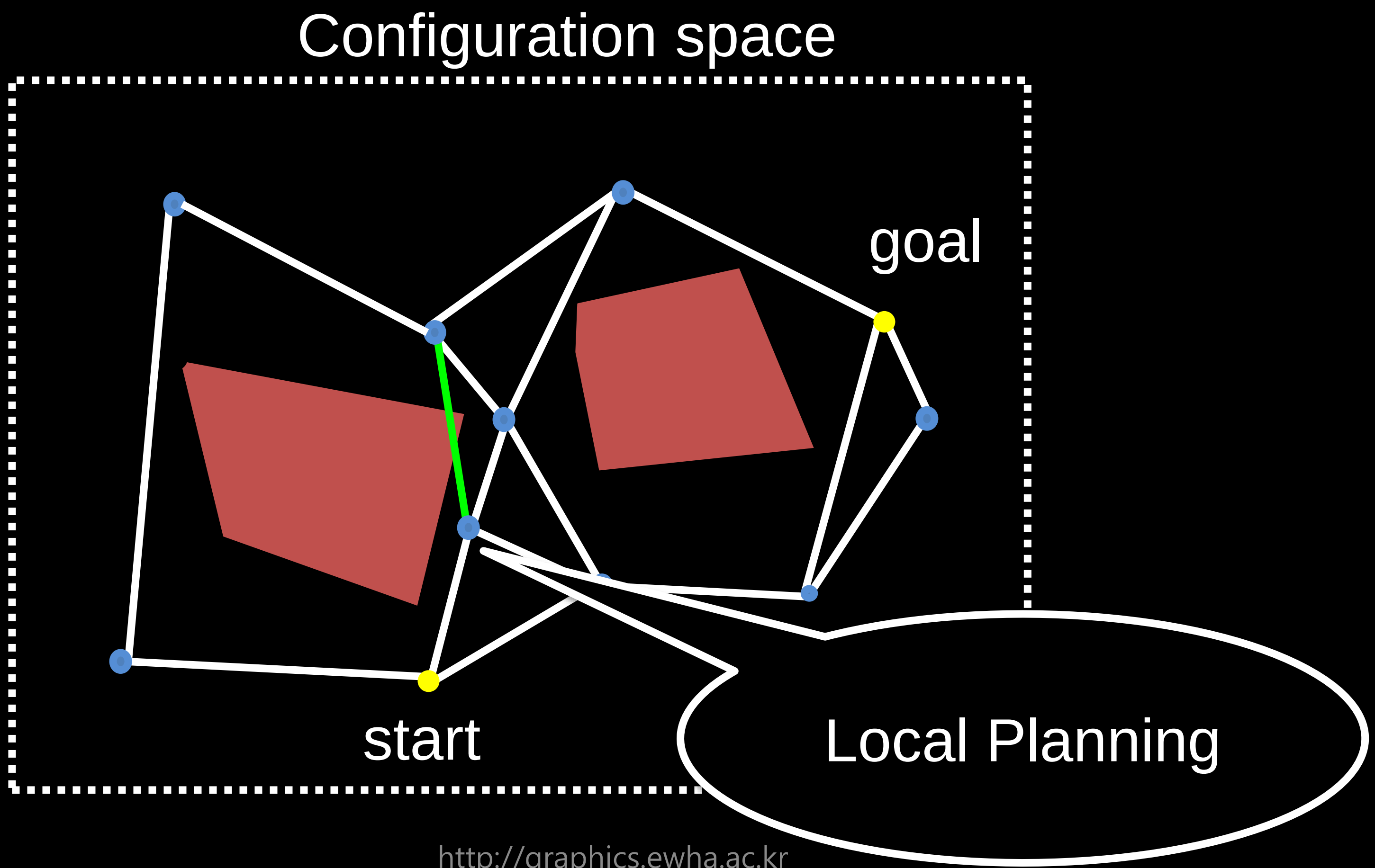
COLLISION CONNECTION QUERY

Problems in Sampling-based Motion Planning

- Completeness
  - Probabilistically-complete
  - Resolution-complete
- Accuracy
- Efficiency

Probabilistic Roadmaps

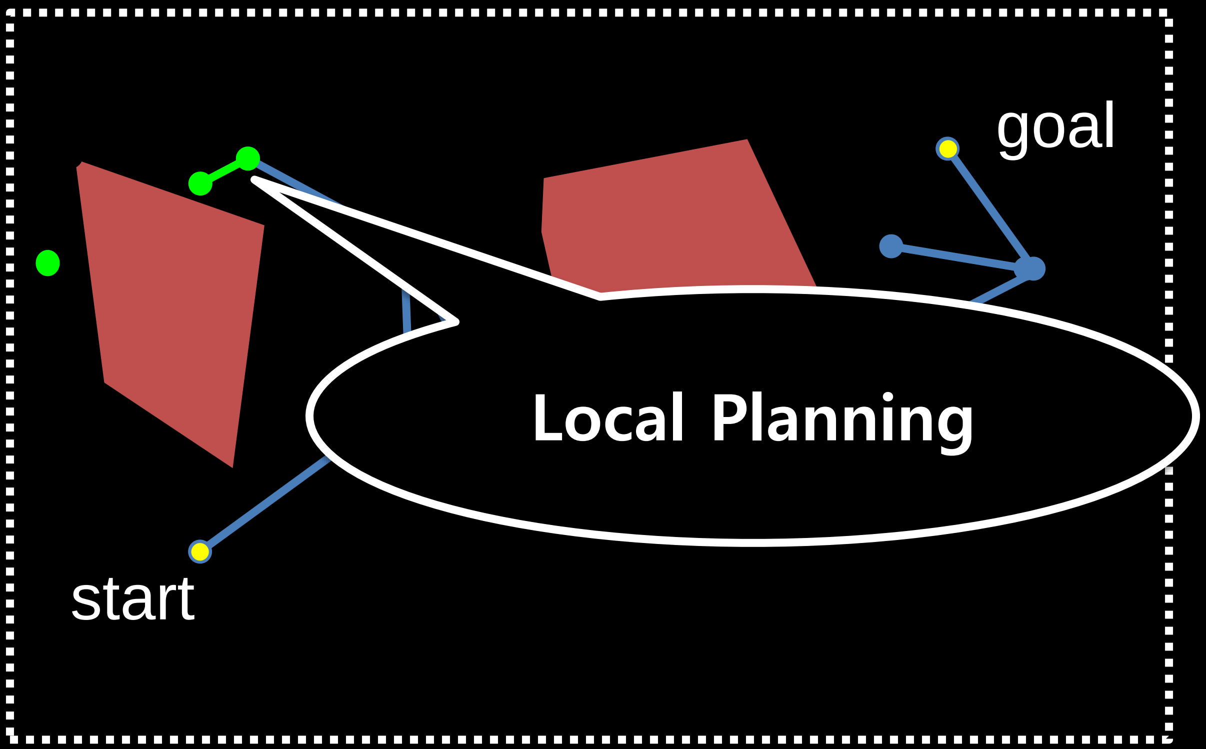
Construction phase and query phase



Rapidly-exploring Random Trees

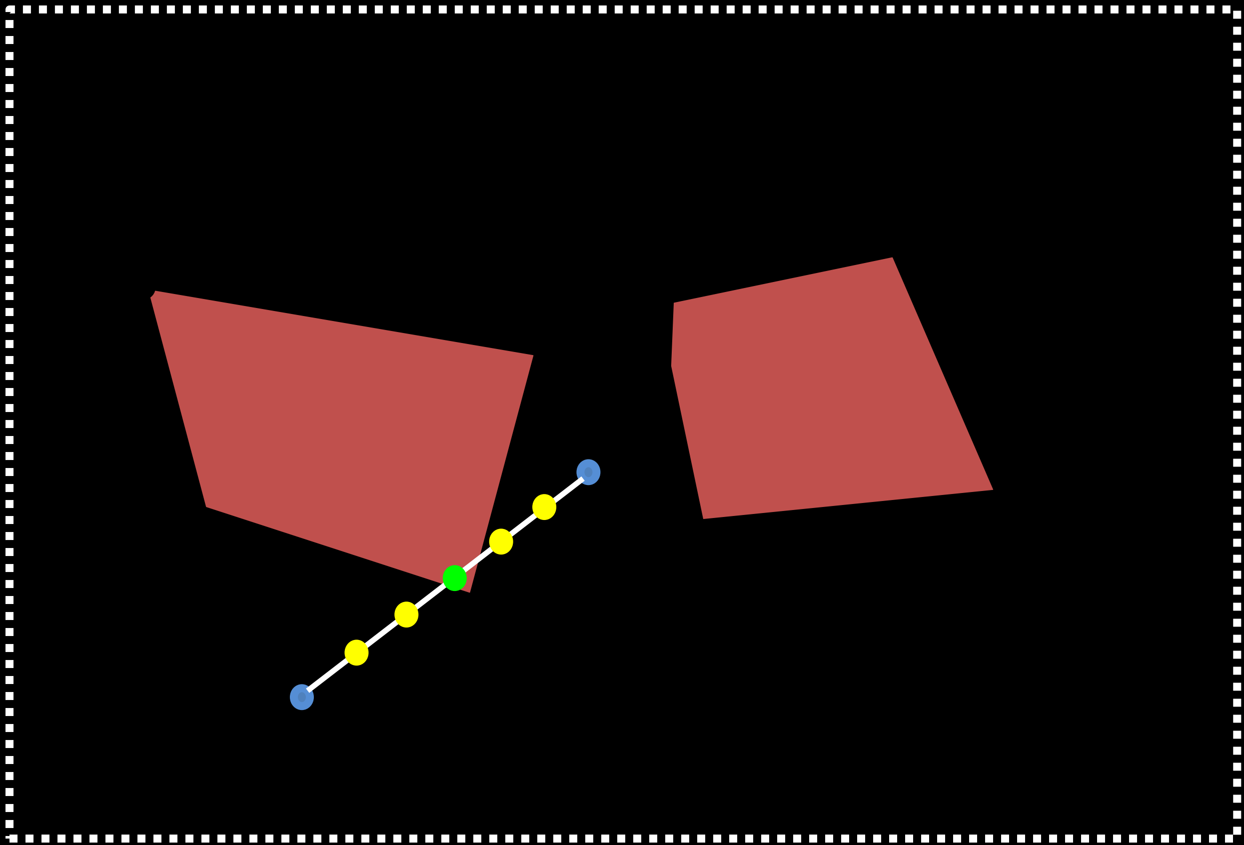
Expansion phase and connect phase

Configuration space

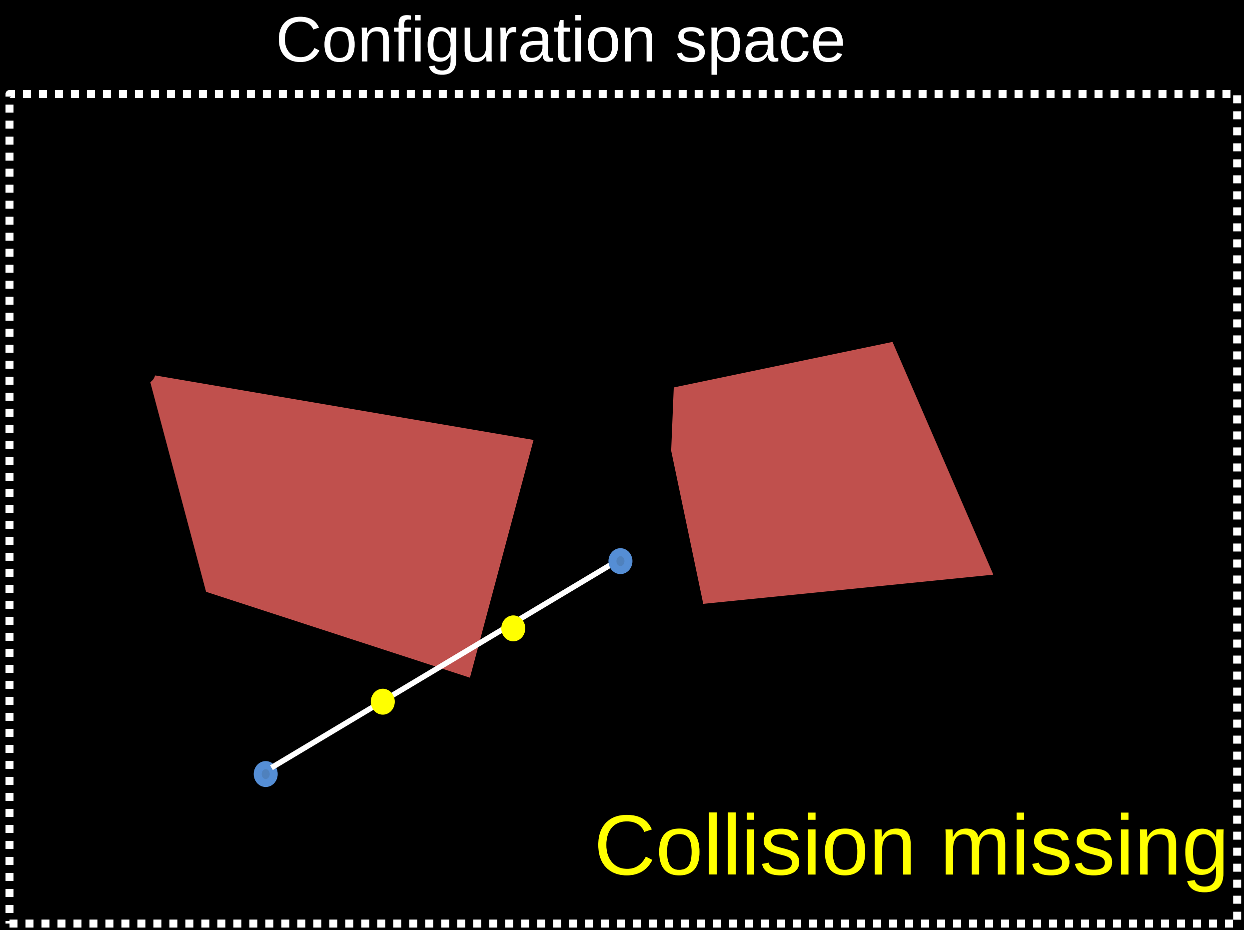


Fixed-resolution Local Planning

Configuration space



Fixed-resolution Local Planning



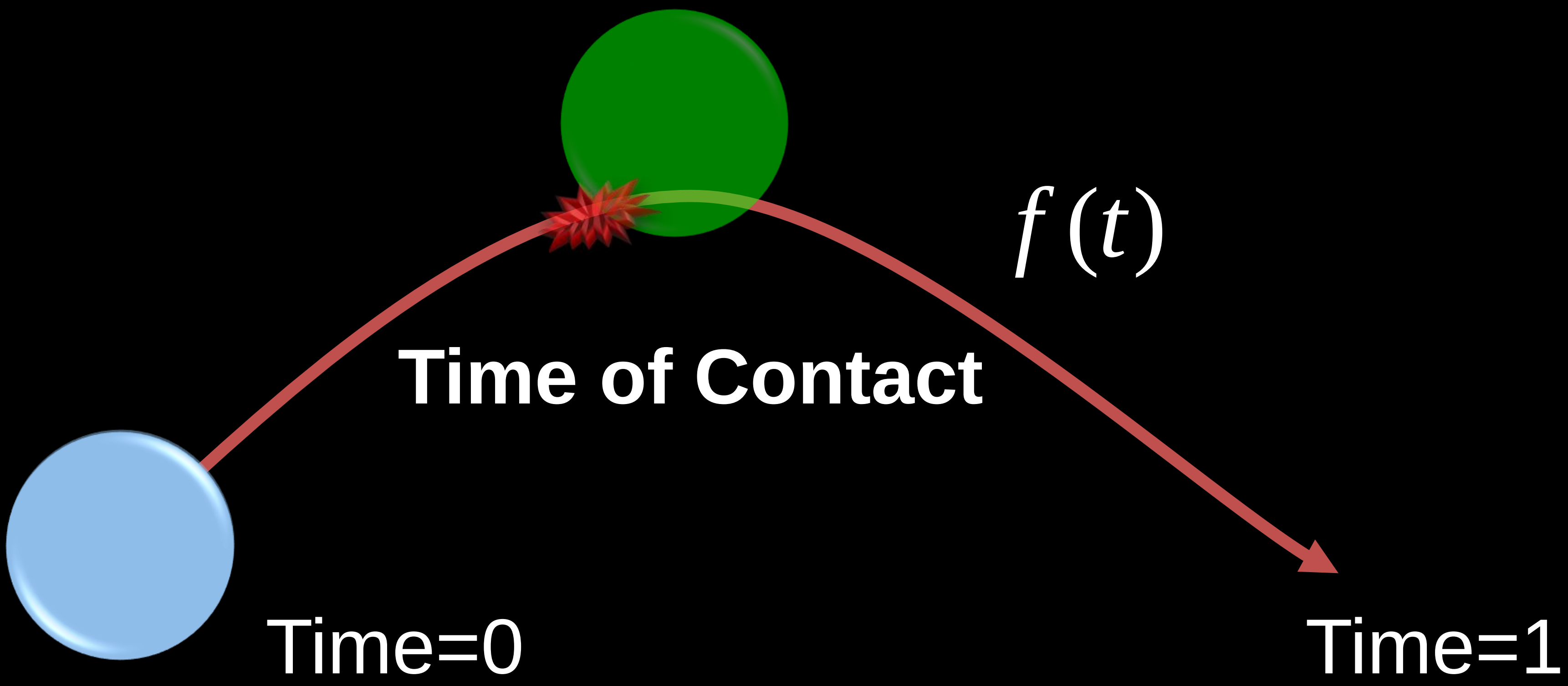
Fixed-resolution Local Planning

- Two problems
  - Collision-miss (accuracy)
  - Collision-resolution (efficiency)

→ Exact collision checking
  - Collision Connection Query (CCQ)

Continuous Collision Detection

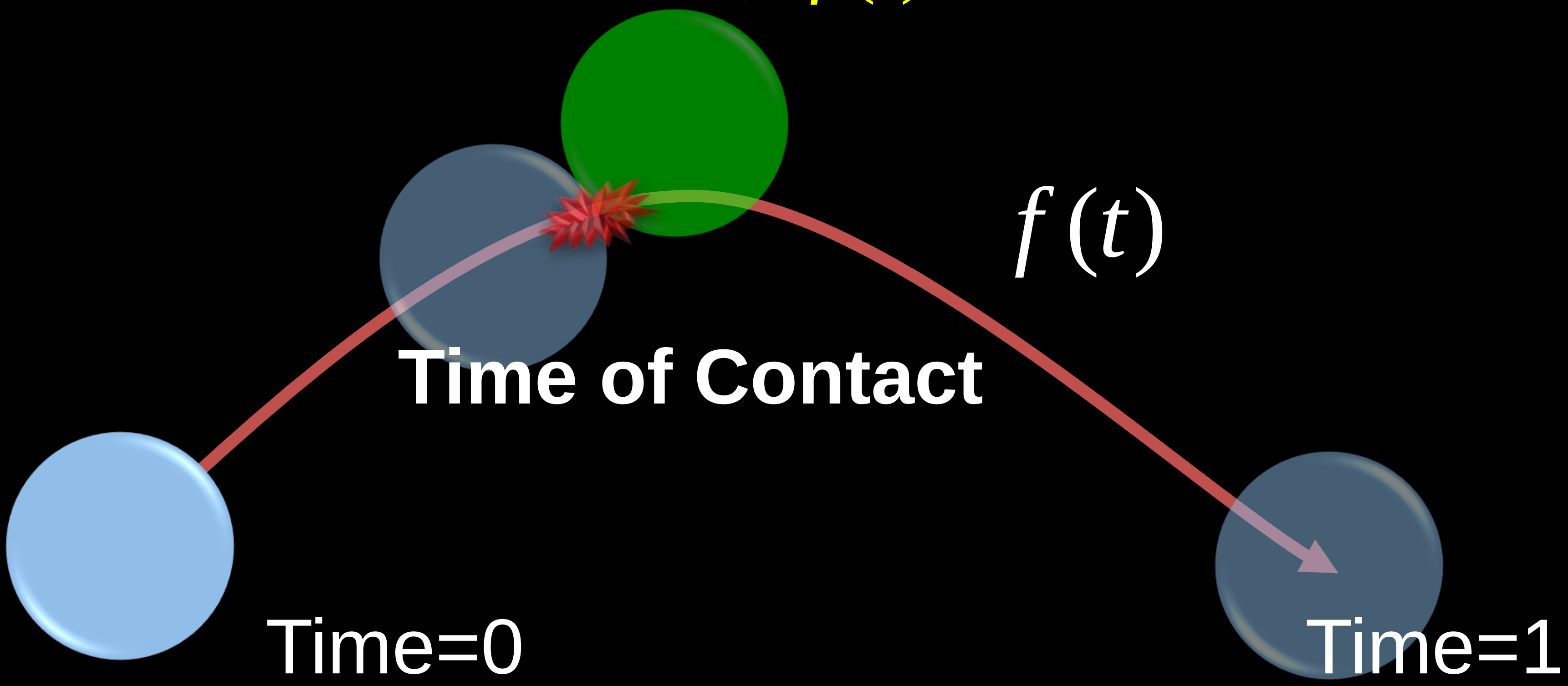
Motion trajectory ***f(t)*** is known in advance



Continuous Collision Detection

- Similar to exact local planning

If Time of Contact < 1, $f(t)$ is in collision



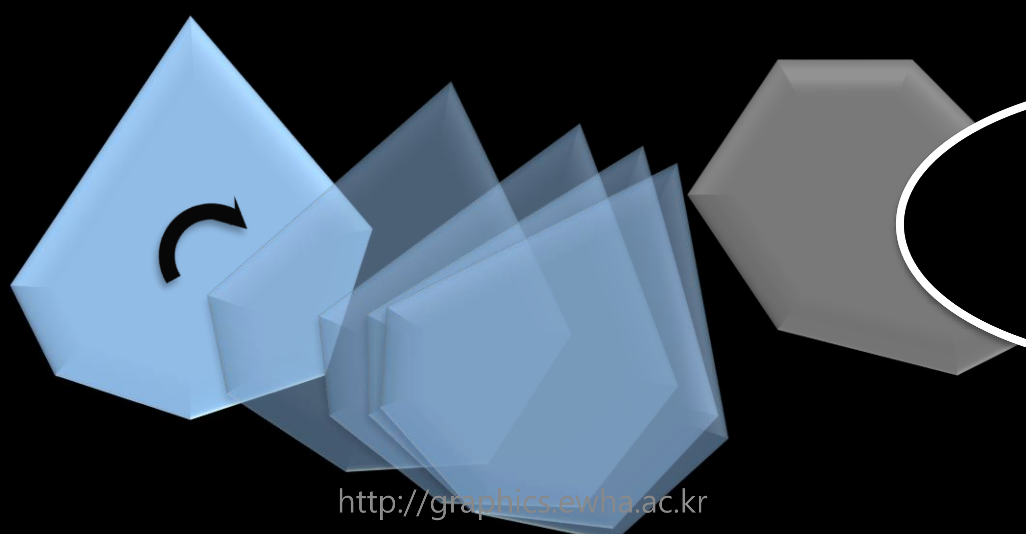
Conservative Advancement (CA)

1. Find a step size Δt_i to conservatively advance the object without collision
2. Repeat until inter-distance < ε

Euclidean Distance

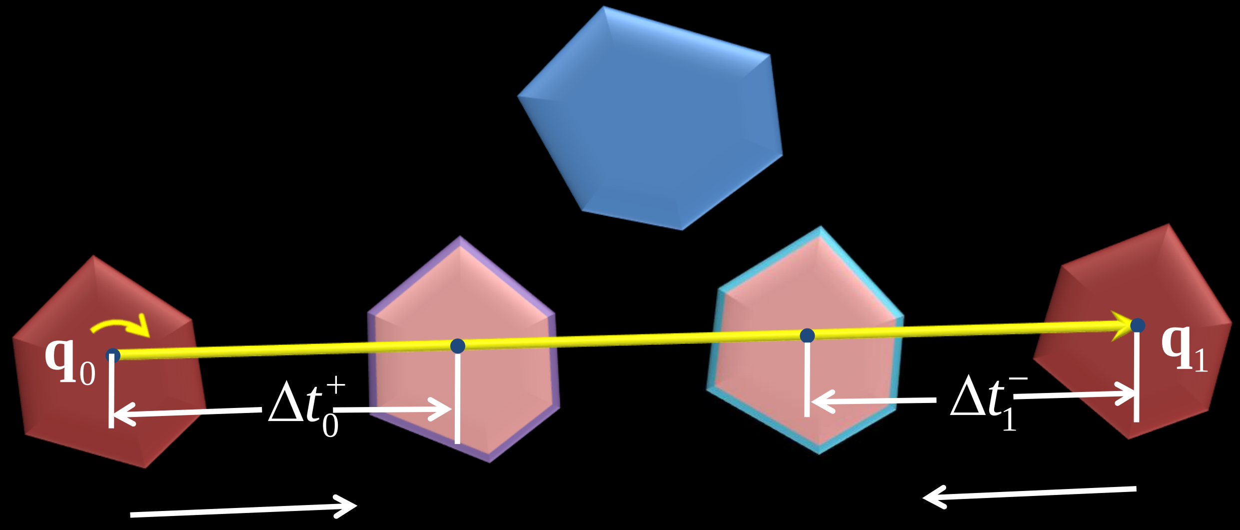
$$TimeofContact = \Delta t_1 + \Delta t_2 + \Delta t_3 + \Delta t_4 \qquad \Delta t_1 \le \frac{d}{\mu}$$

Motion Bound



Boolean CCQ_s Query

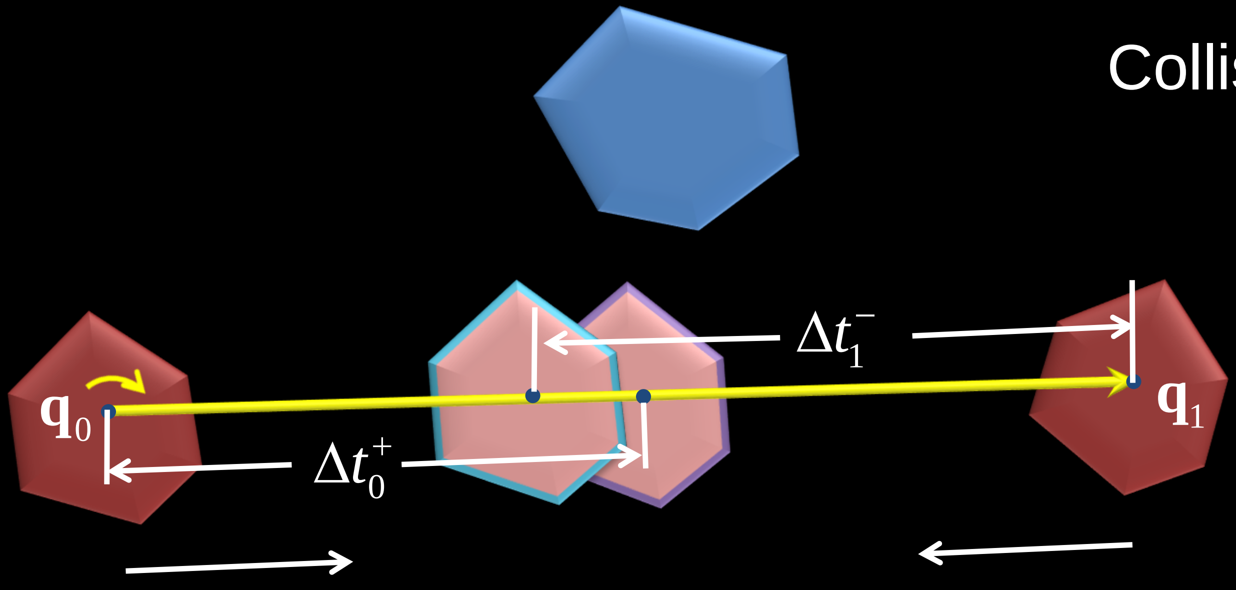
- Dual advancements from the both end-configurations



Boolean CCQ_s Query

$$\Delta t_0^{\ +} + \Delta t_1^{\ -} \geq 1$$

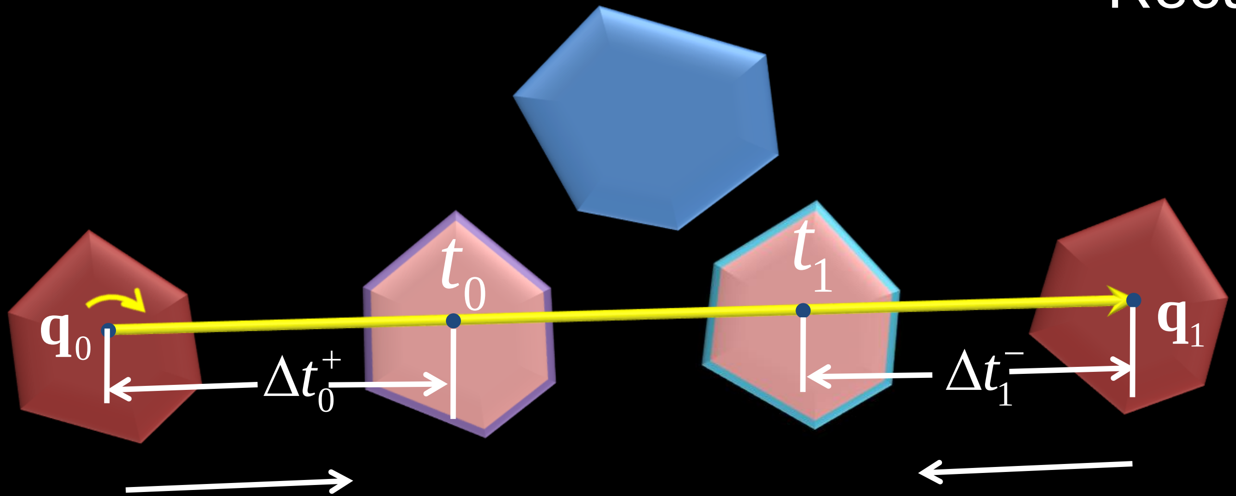
Collision-free



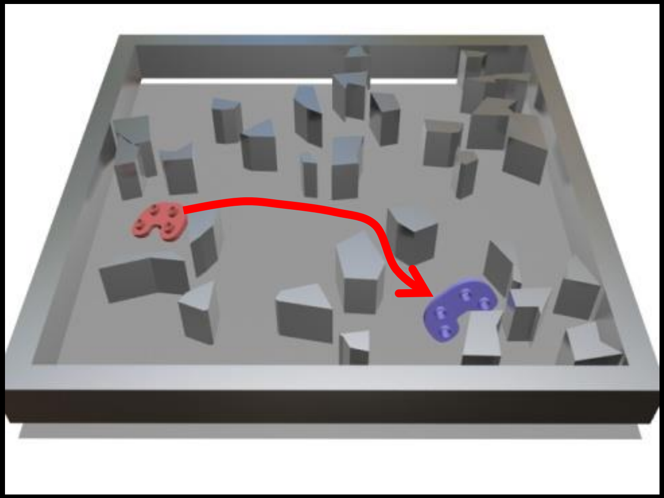
Boolean CCQ_s Query

$$\Delta t_0^{+} + \Delta t_1^{-} < 1$$

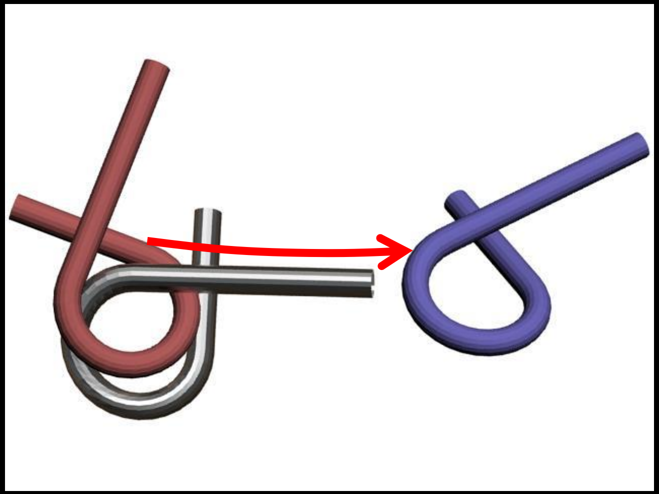
Recurse on $[t_0, t_1]$



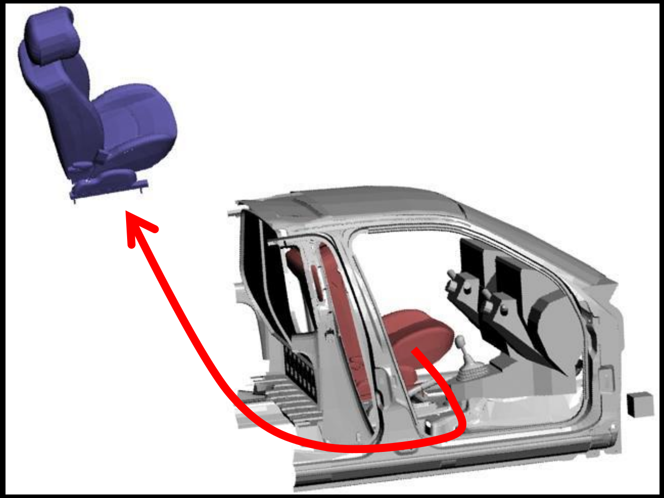
Benchmarking Scenes



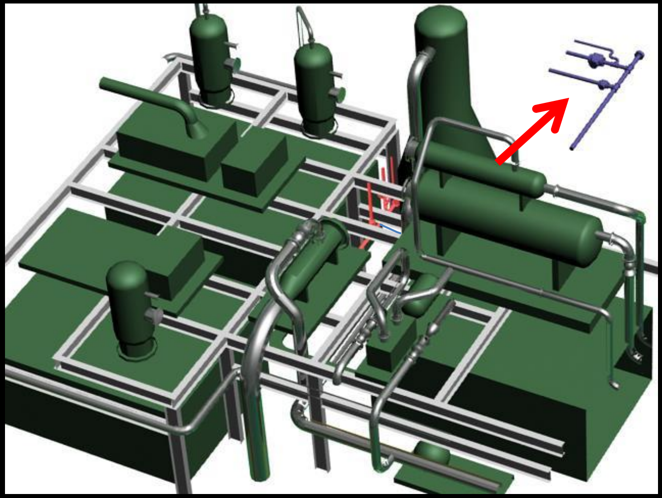
2.5K tris + 0.9K tris



1K tris + 1K tris

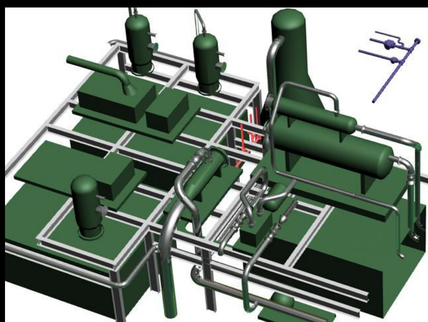


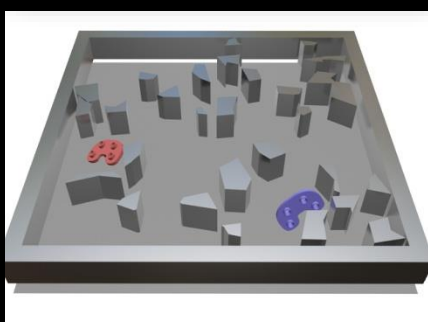
15K tris + 30K tris

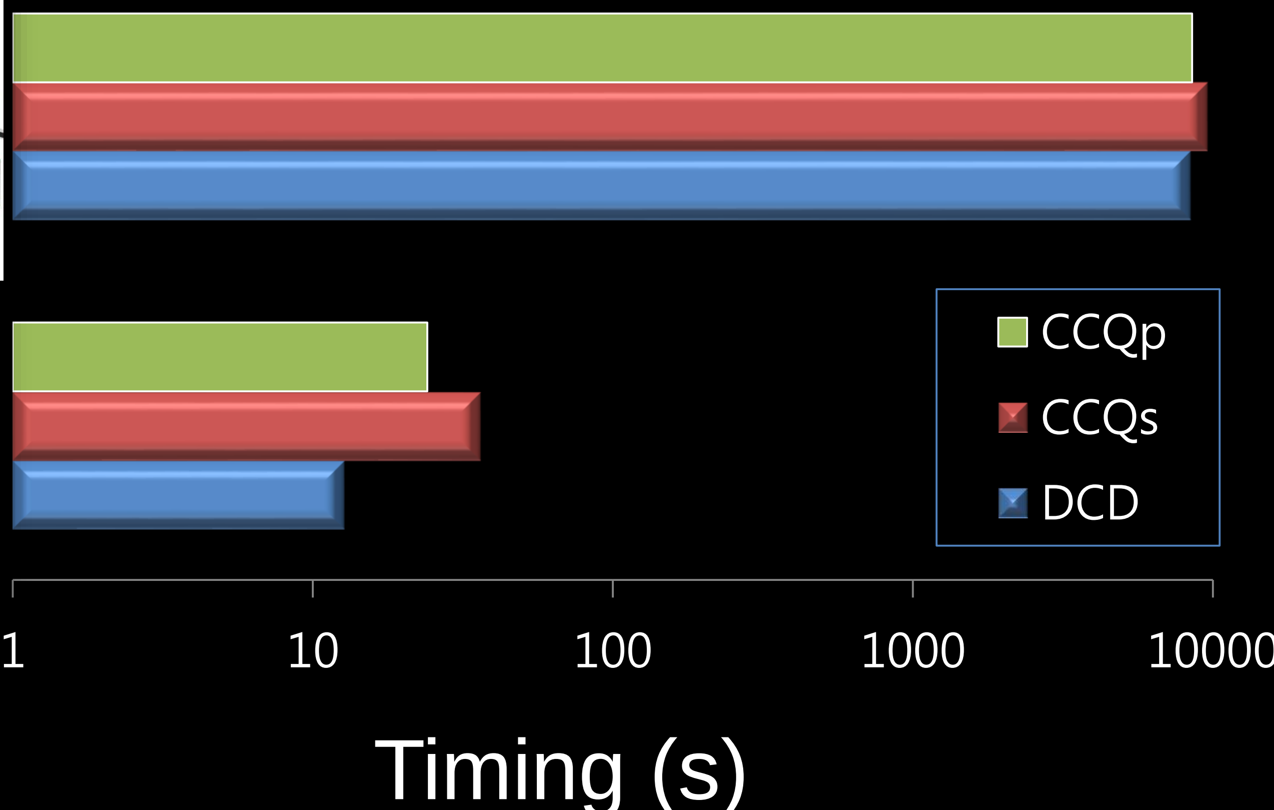


10K tris+ 38K tris

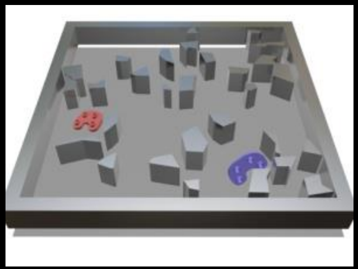
PRM with CCQ

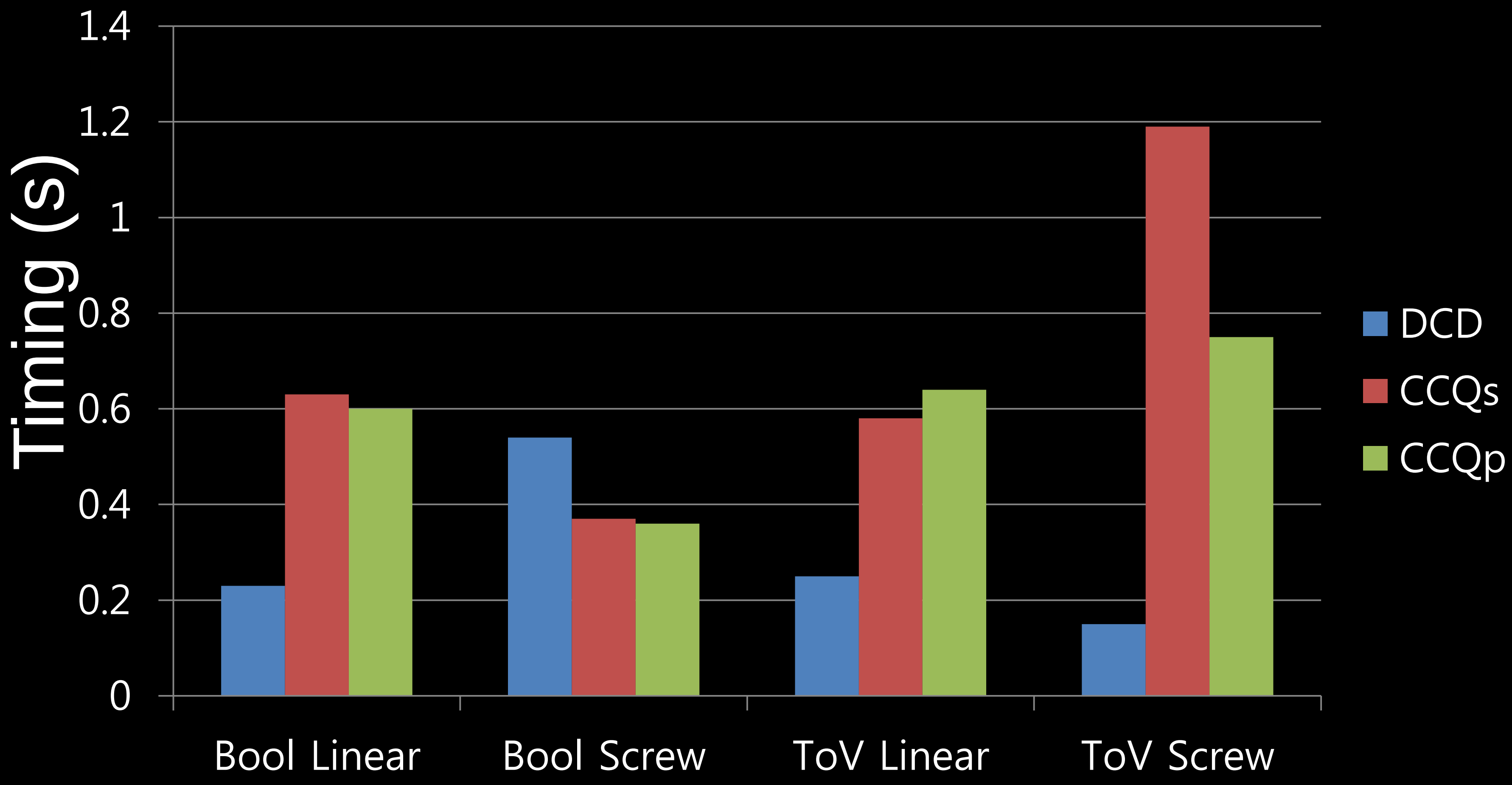




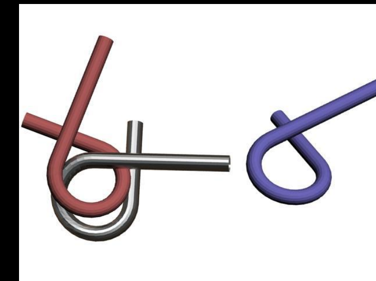


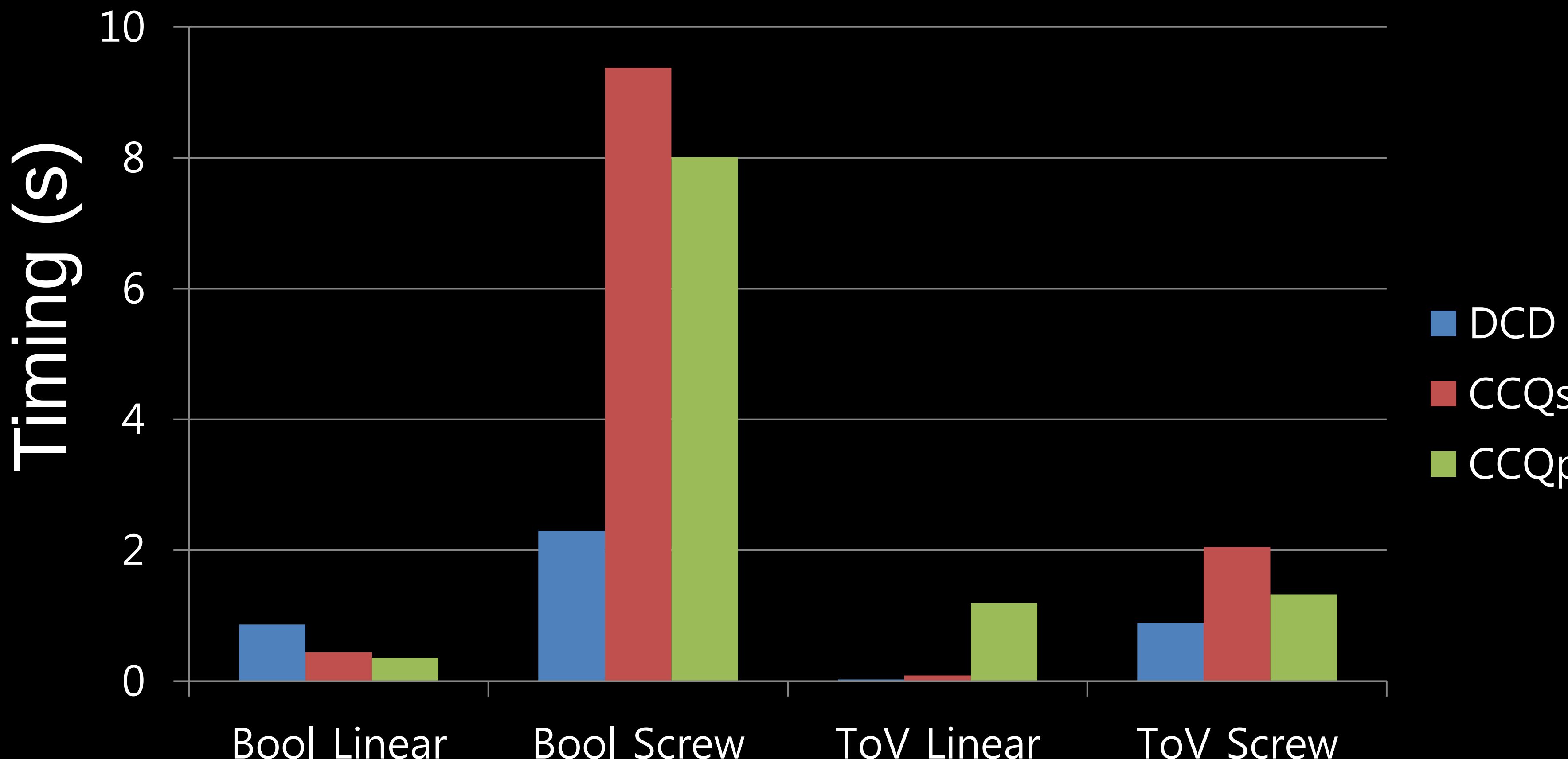
RRT with CCQ



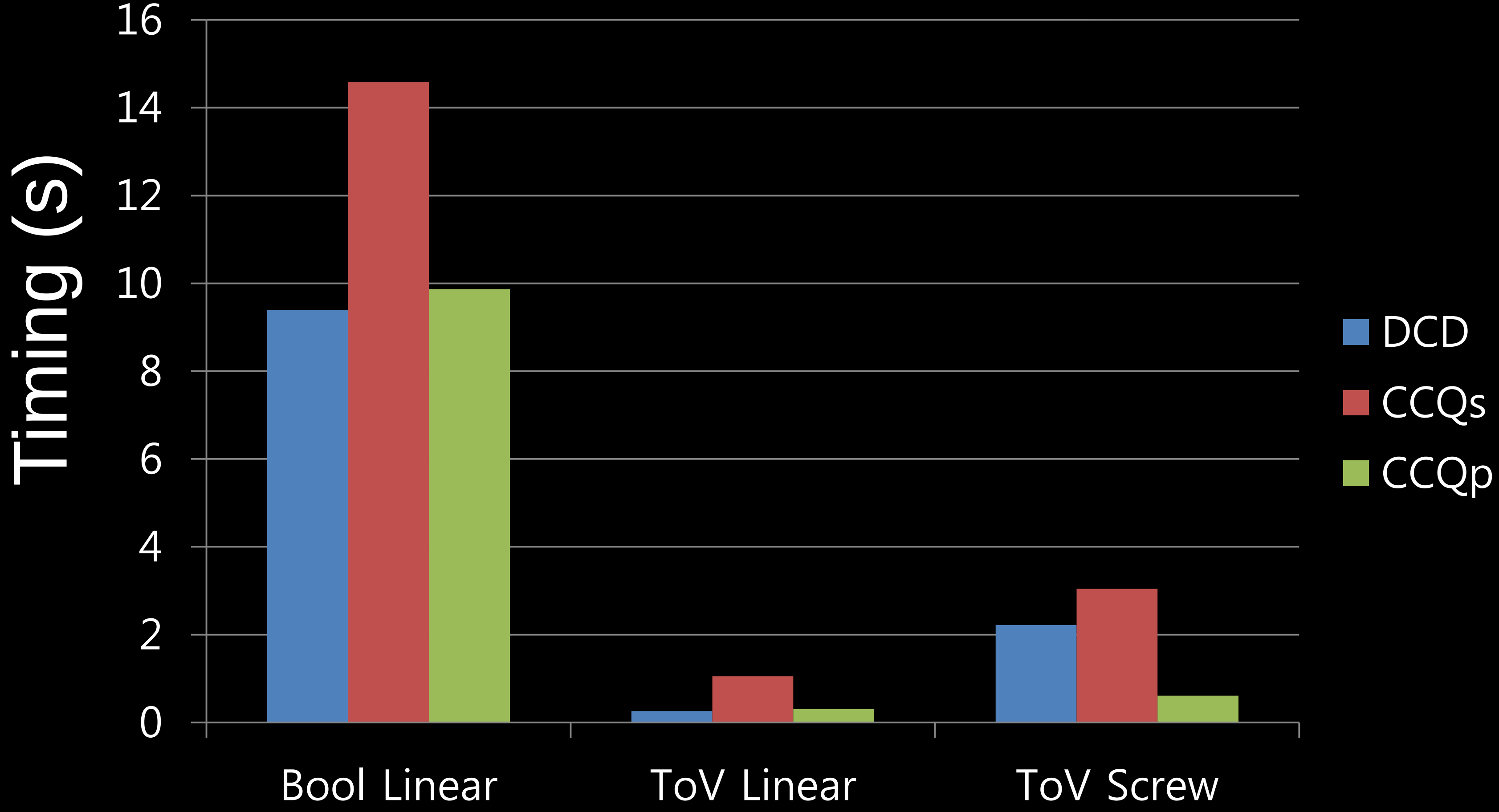

Timing (s)
1.4
1.2
1
0.8
0.6
0.4
0.2
0
Bool Linear
Bool Screw
ToV Linear
ToV Screw
DCD
CCQs
CCQp

RRT with CCQ

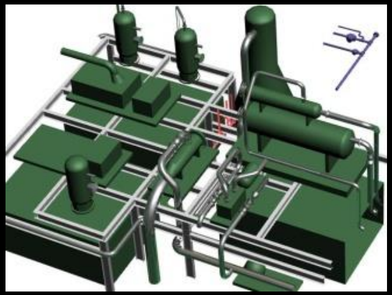


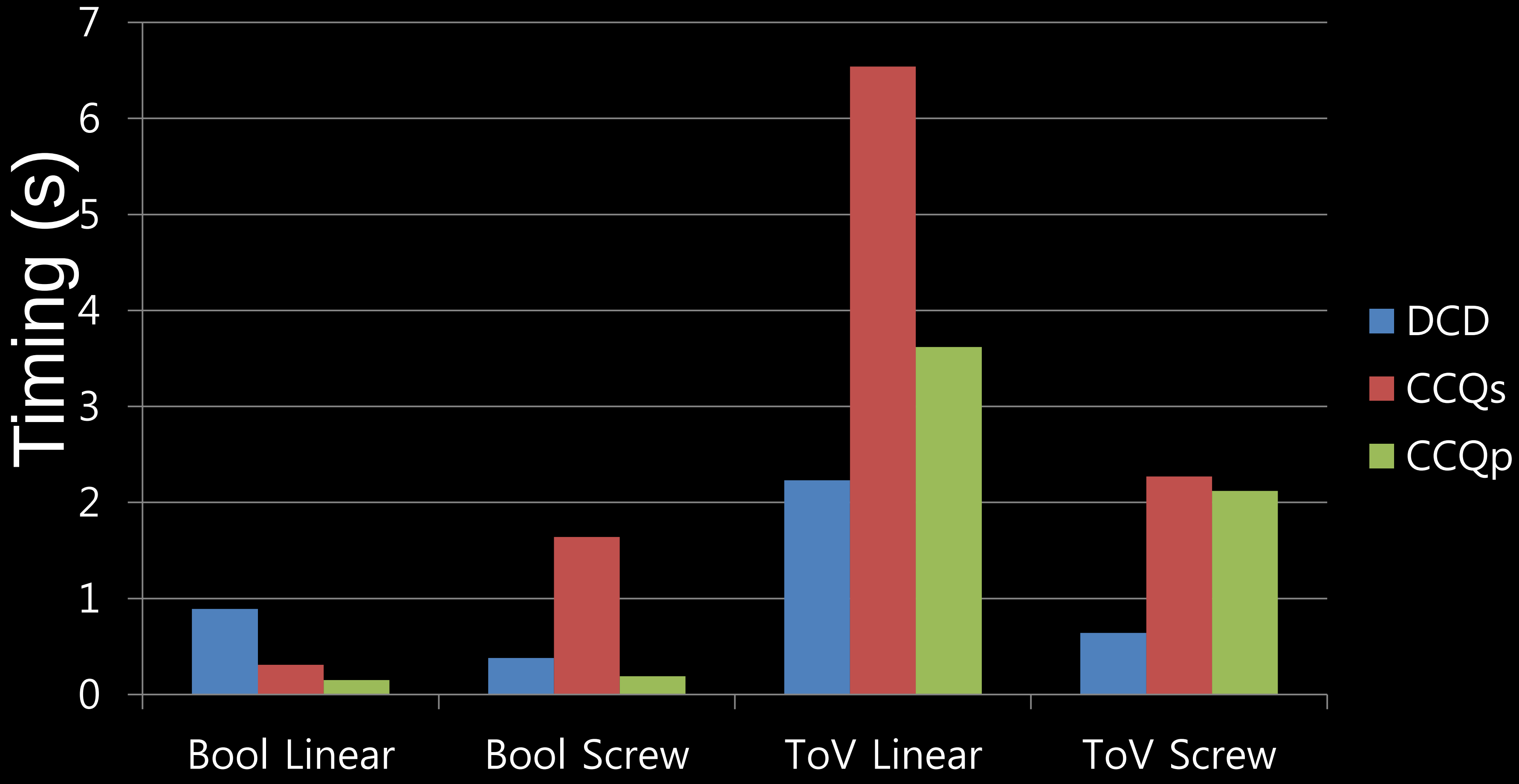

Timing (s)
10
8
6
4
2
0
Bool Linear
Bool Screw
ToV Linear
ToV Screw
DCD
CCQs
CCQp

RRT with CCQ


Timing (s)
16
14
12
10
8
6
4
2
0
Bool Linear
ToV Linear
ToV Screw
DCD
CCQs
CCQp

RRT with CCQ

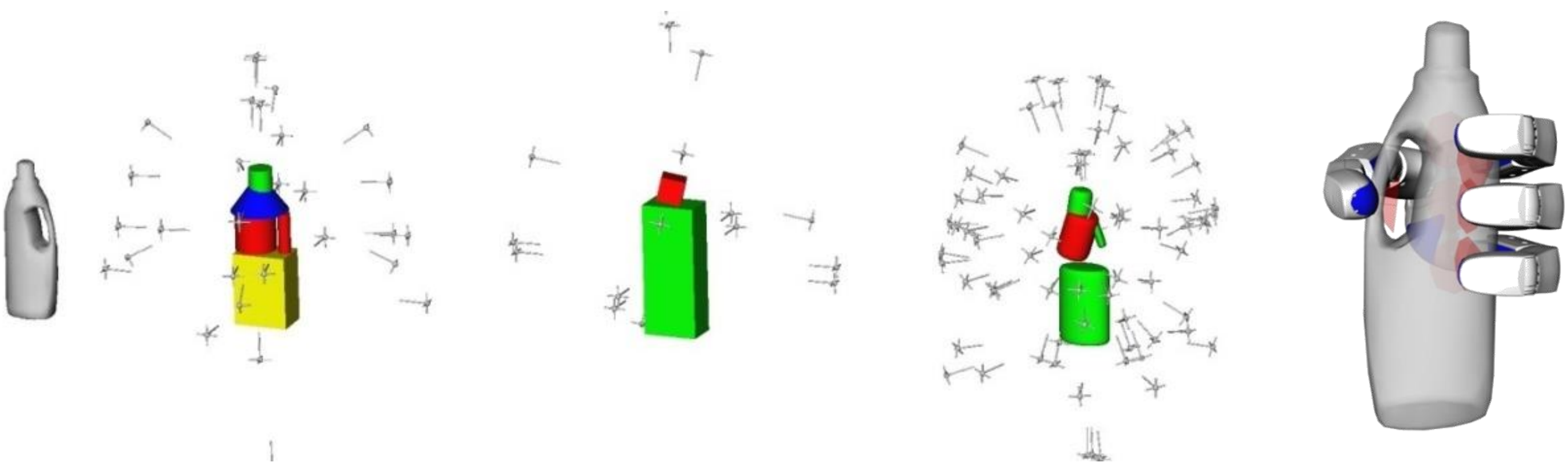




Timing (s)
7
6
5
4
3
2
1
0
Bool Linear
Bool Screw
ToV Linear
ToV Screw
DCD
CCQs
CCQp

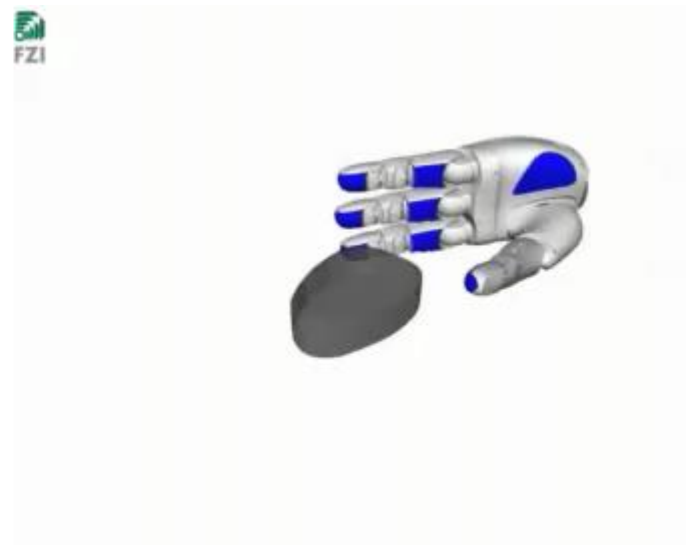
Forward Grasp Planning

1. Generate an approach direction
2. Find contact points
3. Measure the grasp quality
4. Repeat this process

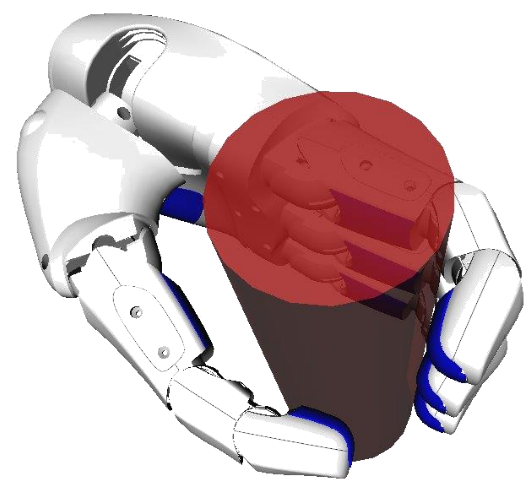


Challenge

- Given grasp approaching directions



- Find all the contact points fast and reliably



Real Robot Execution

[Courtesy of Zhixing Xue]

@KRoC 2014

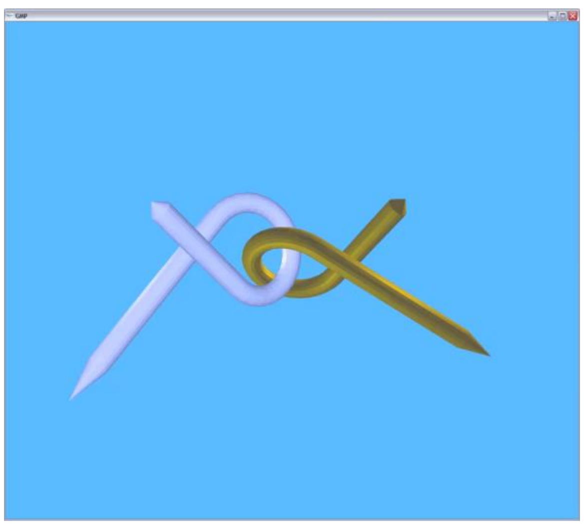
TC1-13 김여진, 단속적 충돌검사를 이용한 효율적 침투깊이 계산 알고리즘

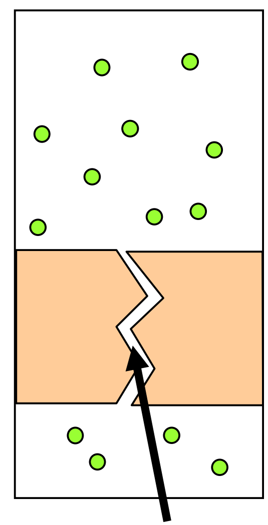
FE1-34 이영은, 다각형 로봇 모델을 위한 연속부호거리 계산 알고리즘

PENETRATION QUERY

Narrow Passage Problem

- Robot needs to operate in cluttered environment

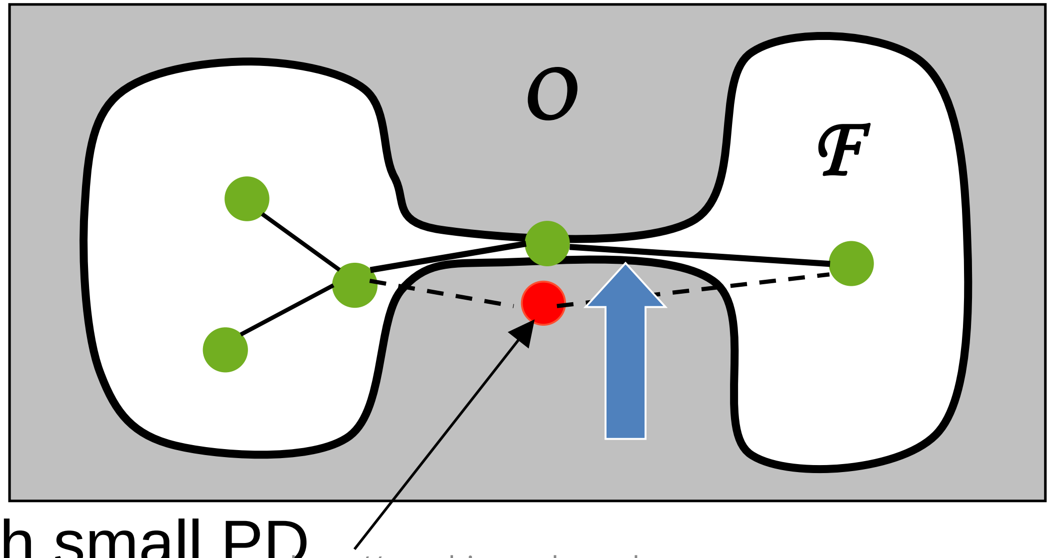




Narrow passage

Retraction-based Planning

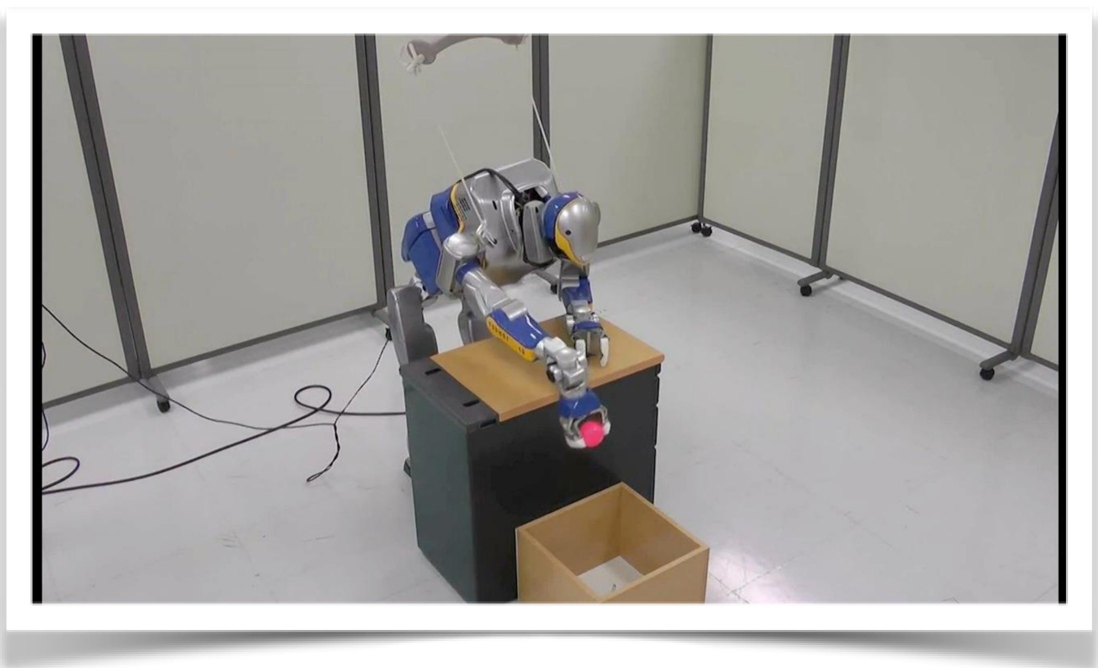
- During roadmap construction, allow milestones with a small PD
- Dilate the free space [Zhang et al. 08]



Milestone with small PD

Optimization-based Motion Planning

- Finds the best joint trajectories that minimize a cost function and satisfy constraints



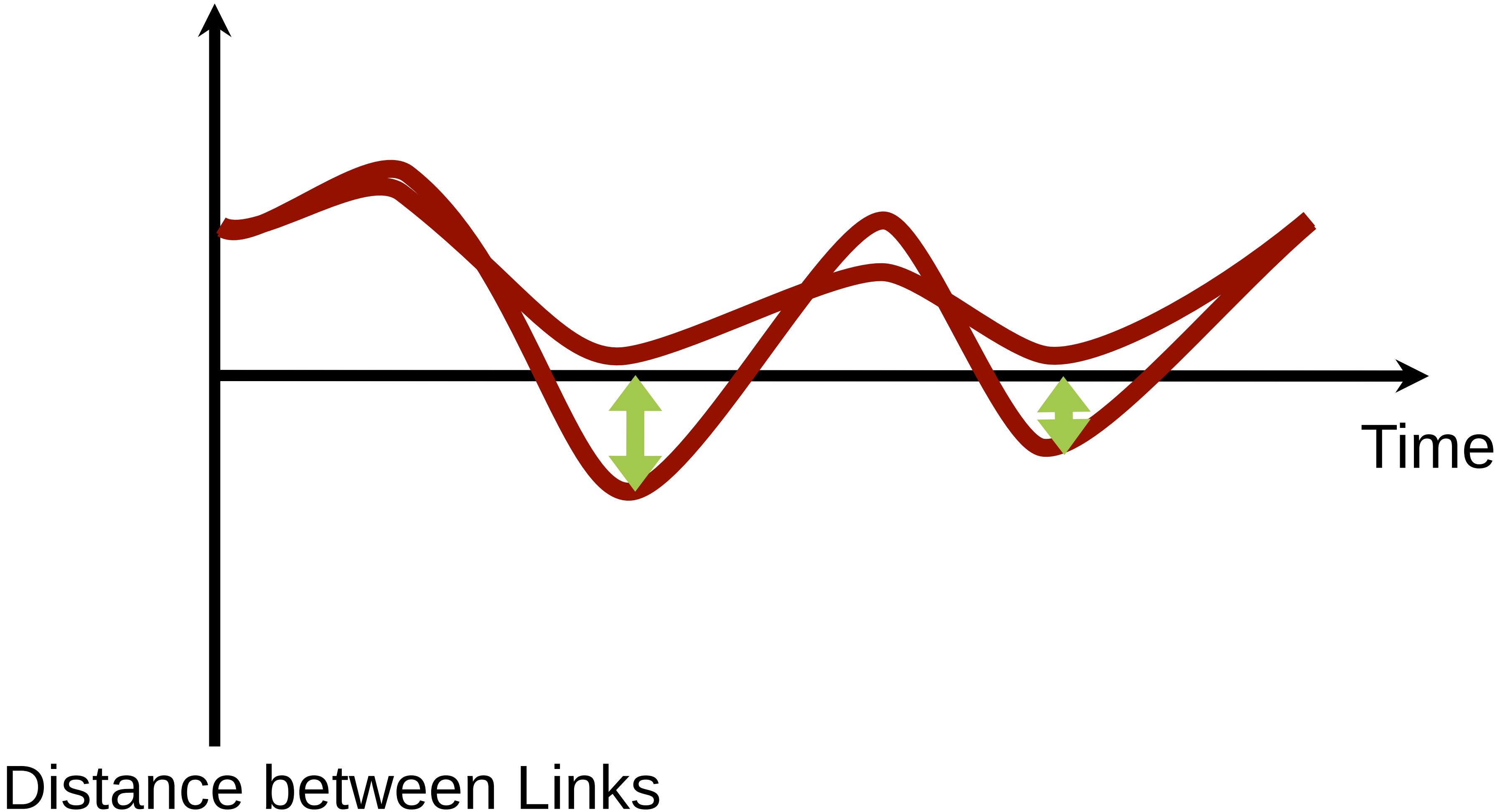
Non-penetration Constraint

- Collision avoidance
- Non-penetration constraint

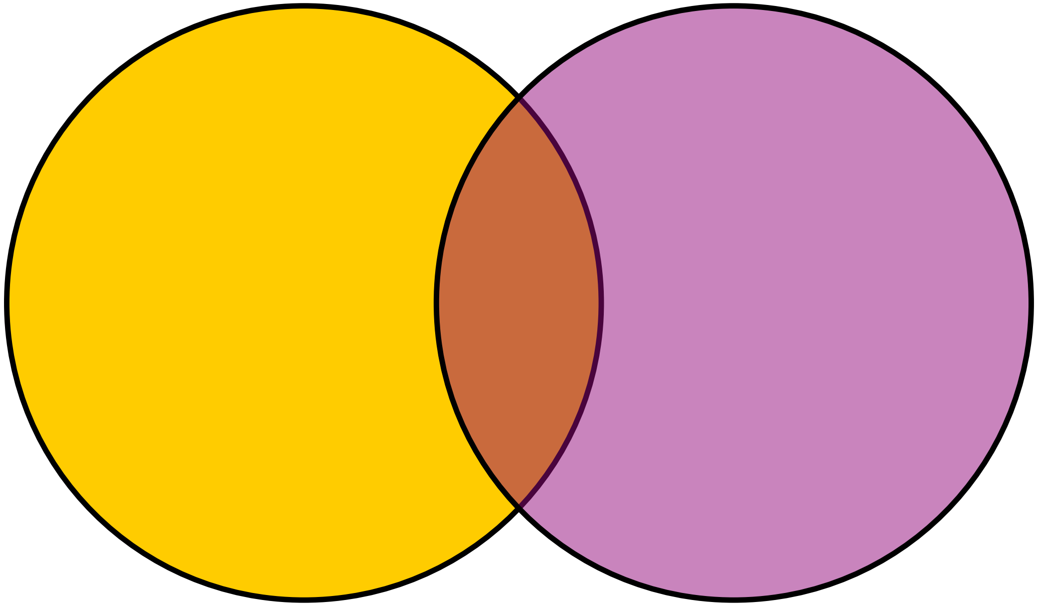
$$\delta(C_i(t), C_j(t)) - \varepsilon \geq 0 \quad \forall t \in [0, T_f]$$

$\delta(\cdot)$: distance function
C_i, C_j: robot's links

Non-penetration Constraint

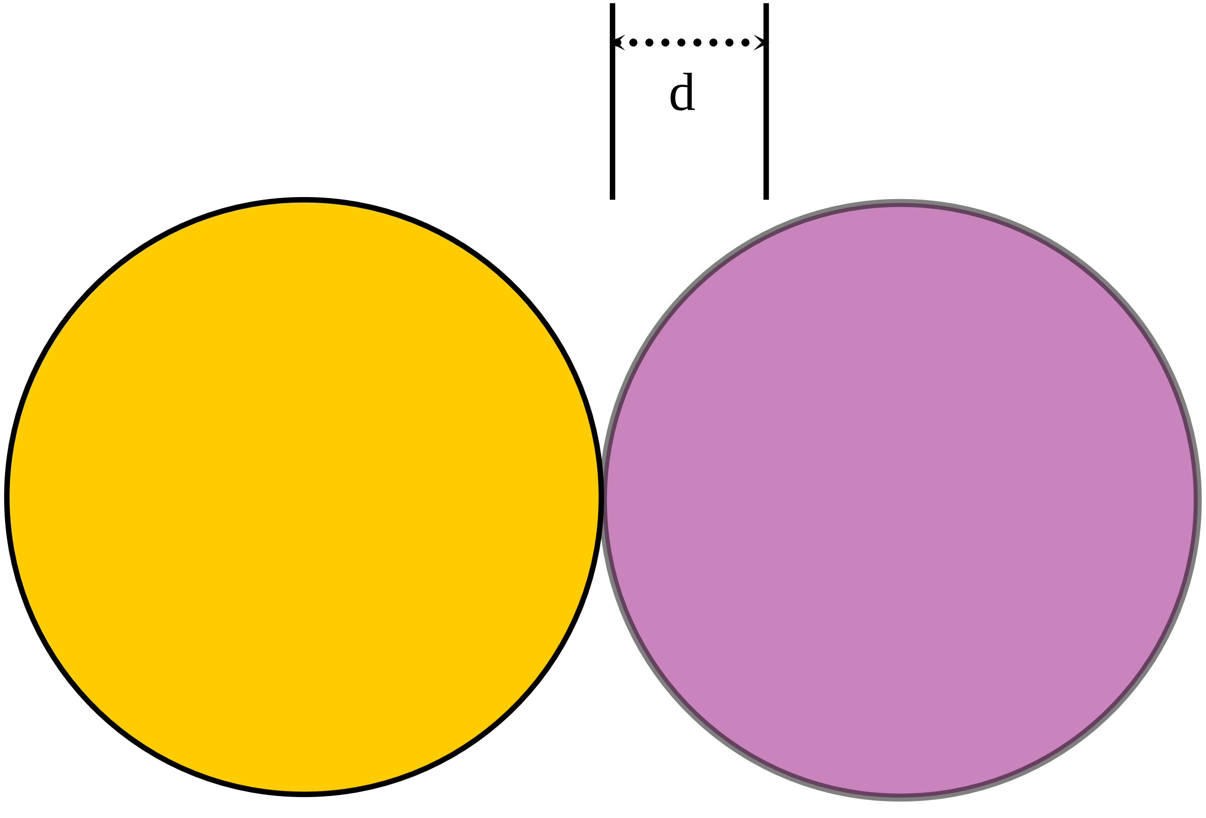


Penetration Depth



Minimum distance needed to separate objects

Penetration Depth

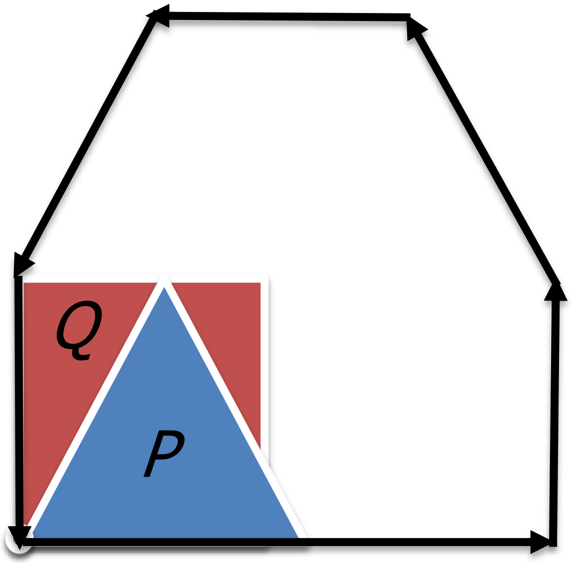


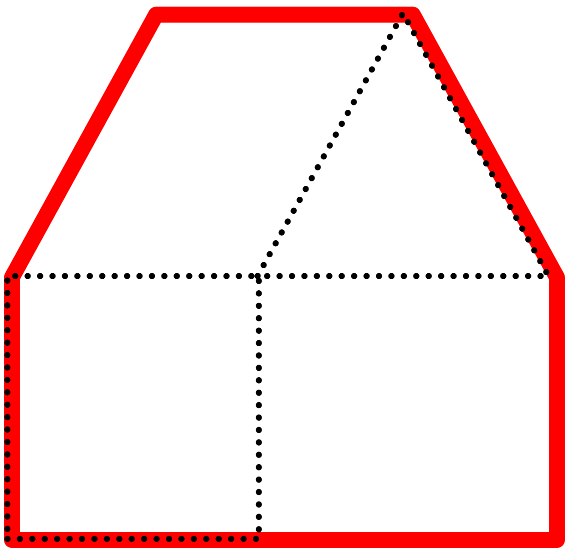
Minimum distance needed to separate objects

Minkowski Sum

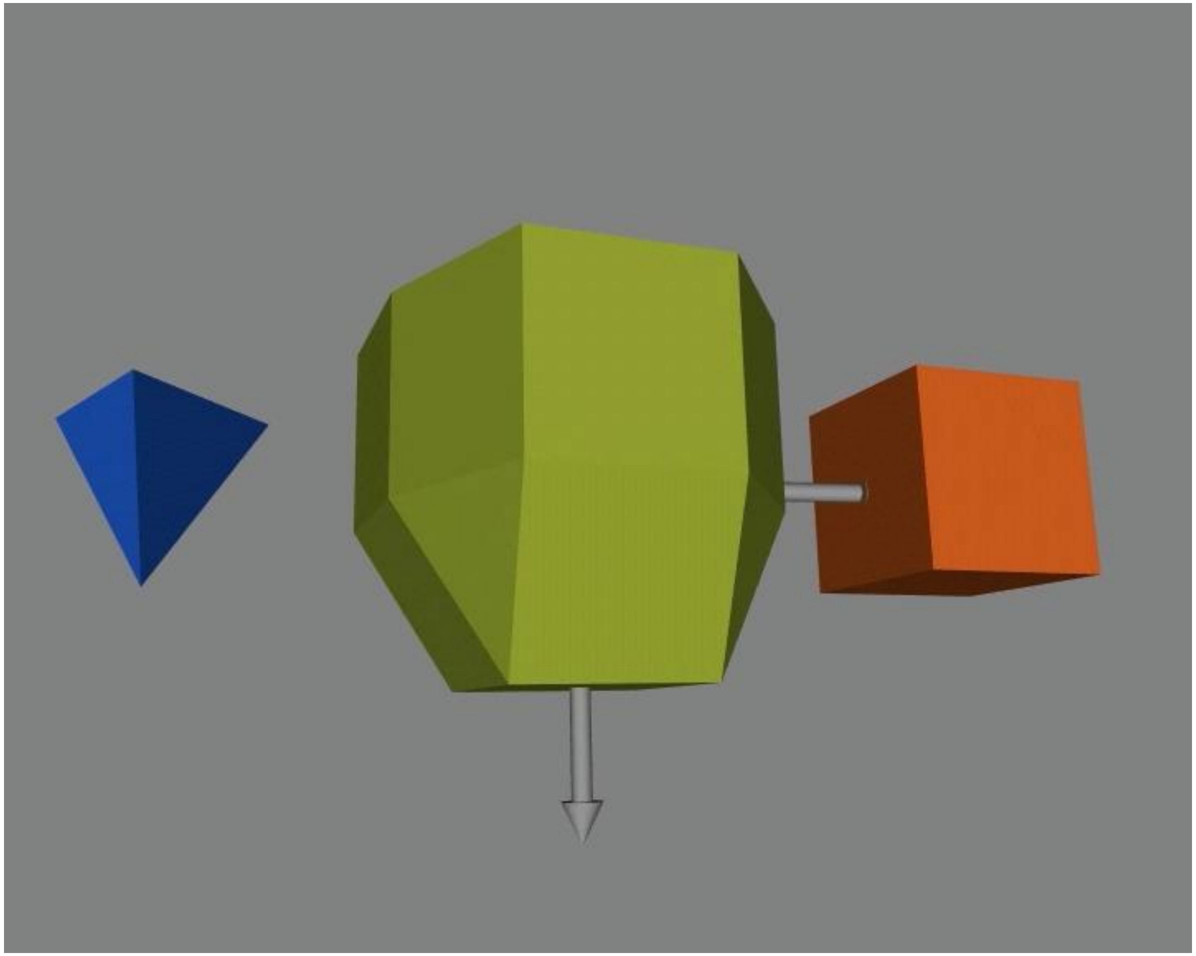
$$P \oplus Q = \{\mathbf{p} + \mathbf{q} \mid \mathbf{p} \in P, \mathbf{q} \in Q\}$$

$$P \oplus -Q = \{\mathbf{p} - \mathbf{q} \mid \mathbf{p} \in P, \mathbf{q} \in Q\}$$

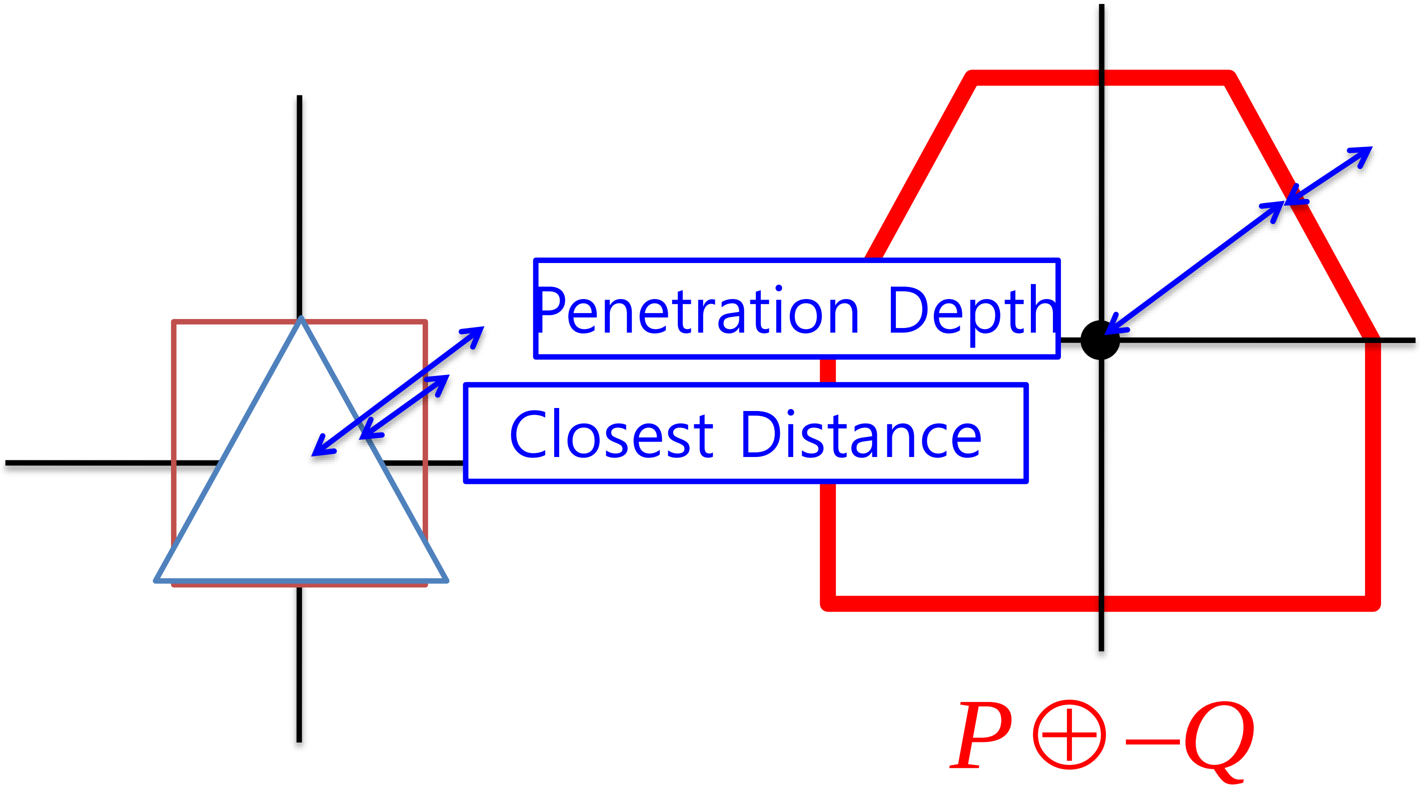




Example

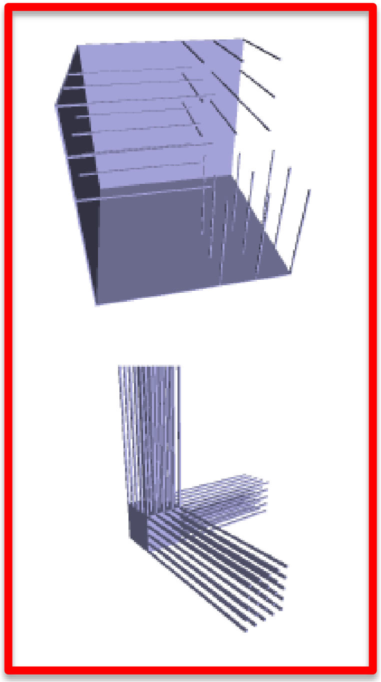


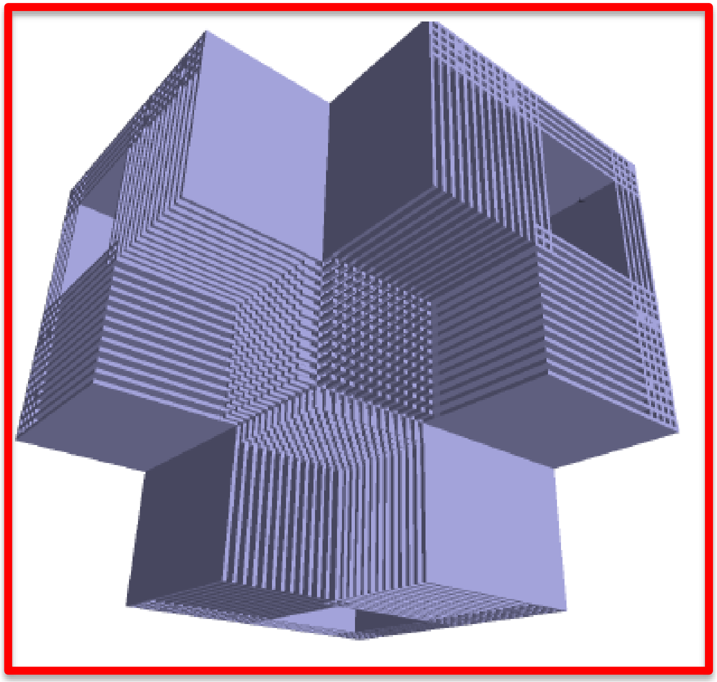
Proximity VS Minkowski Sum



Combinatorial Explosion

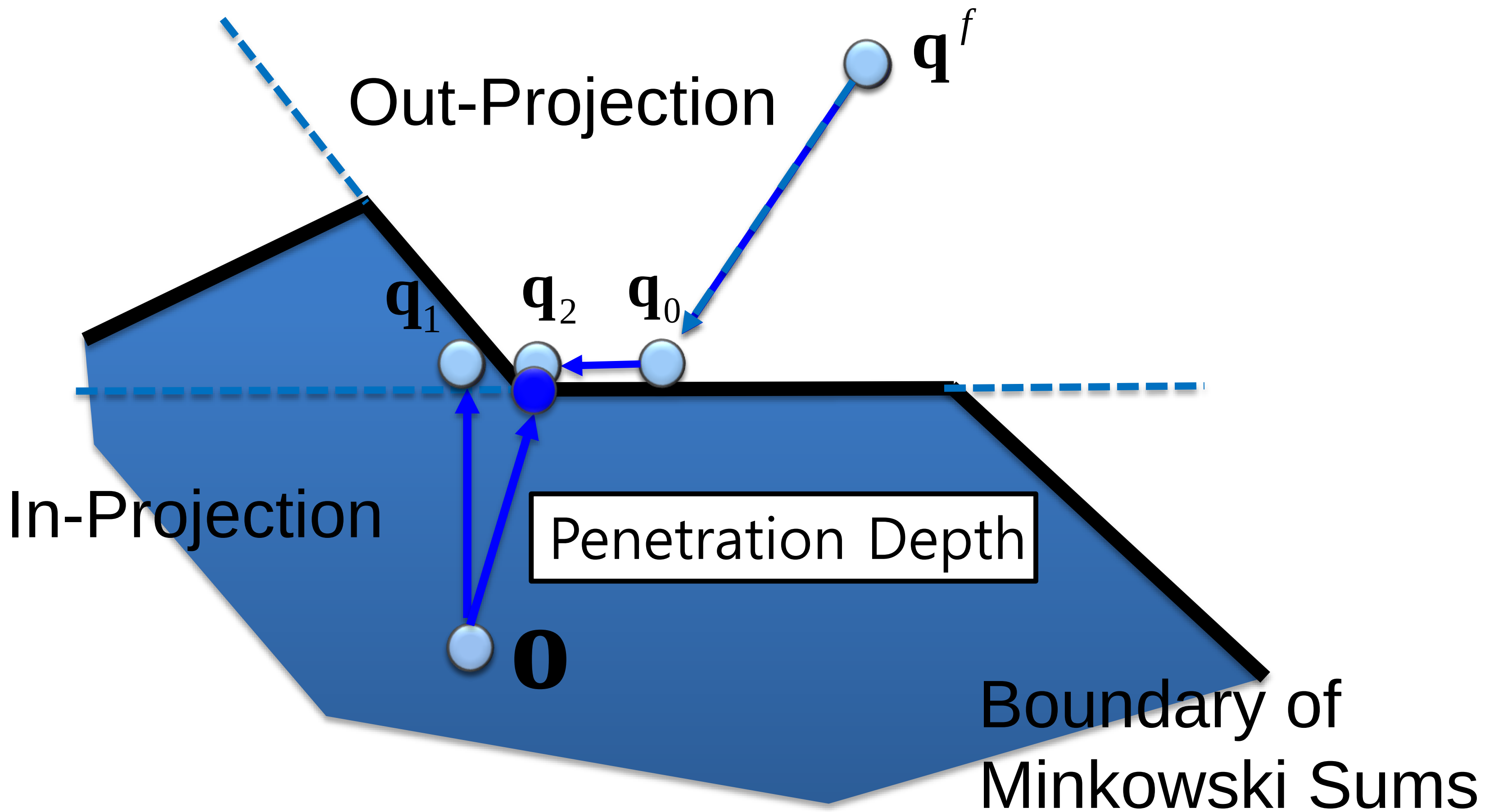
- Complexity of Minkowski Sum
  - $O(m^3n^3)$ with m and n triangles



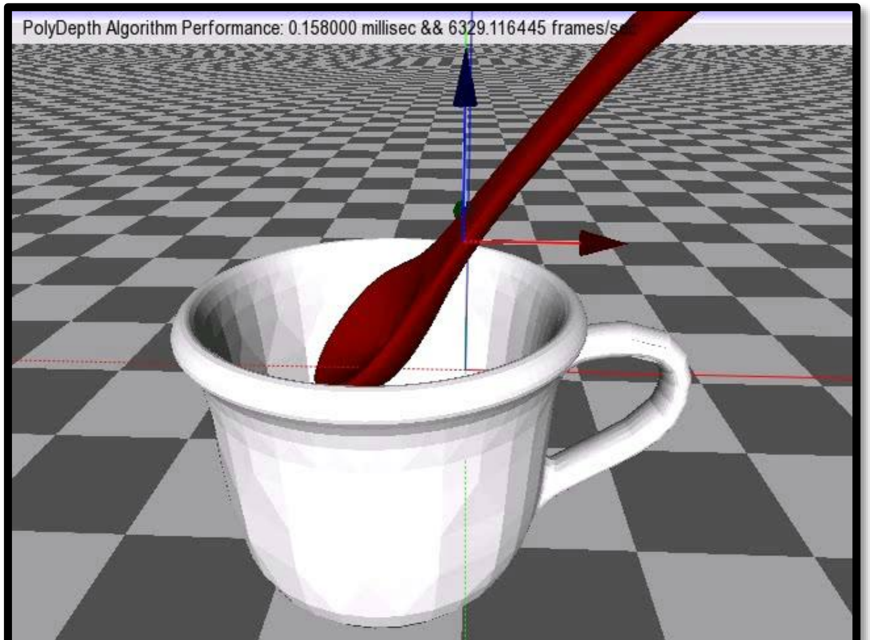


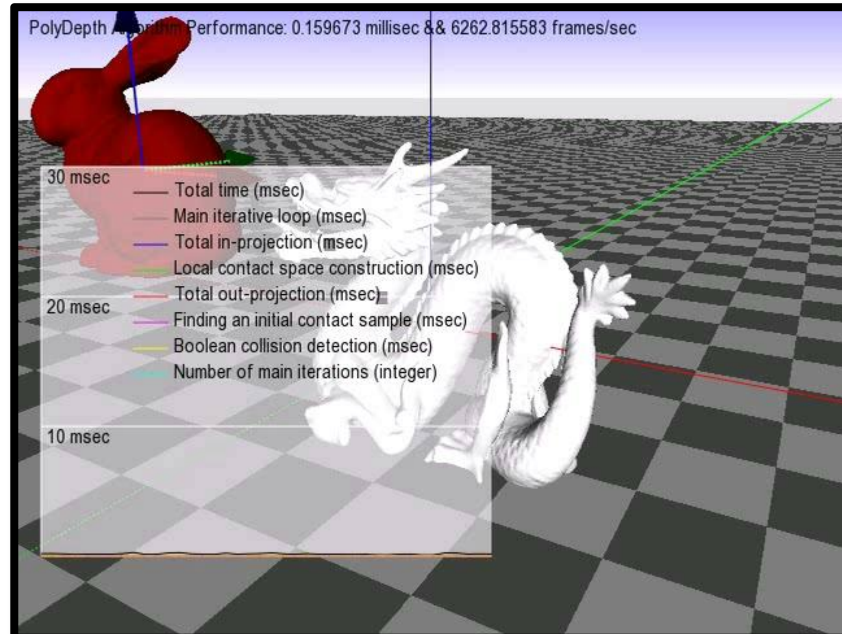
PolyDepth: Iterative Optimization

Je et. al, ACM Transactions on Graphics 2012



PolyDepth Performance



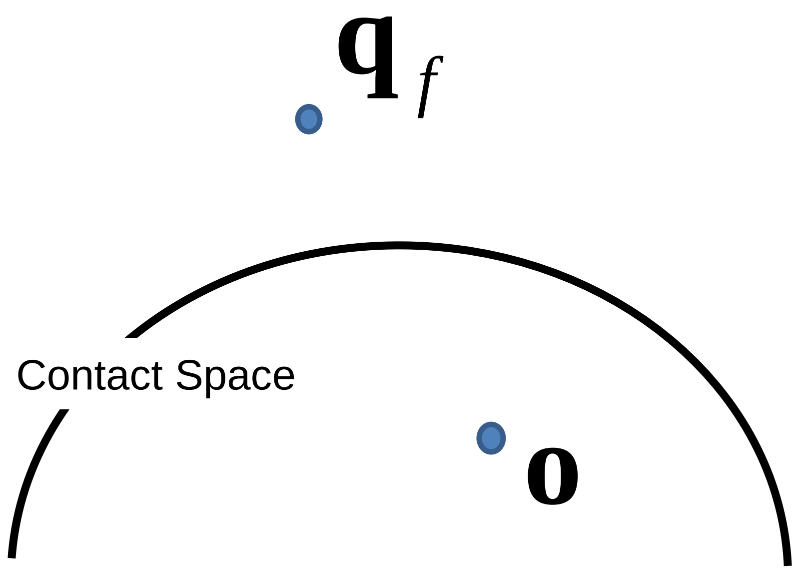


Spoon: 1.3K triangles
Cup: 8.4K triangles
Time: 1~7 msec

Bunny: 40K triangles
Dragon: 174K triangles
Time: 2~15 msec

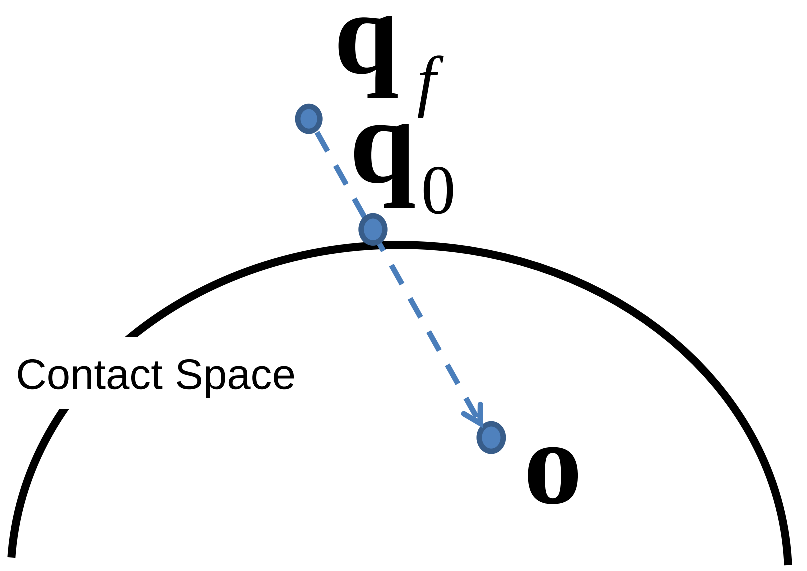
PolyDepth++ Algorithm

1. Free-configuration selection



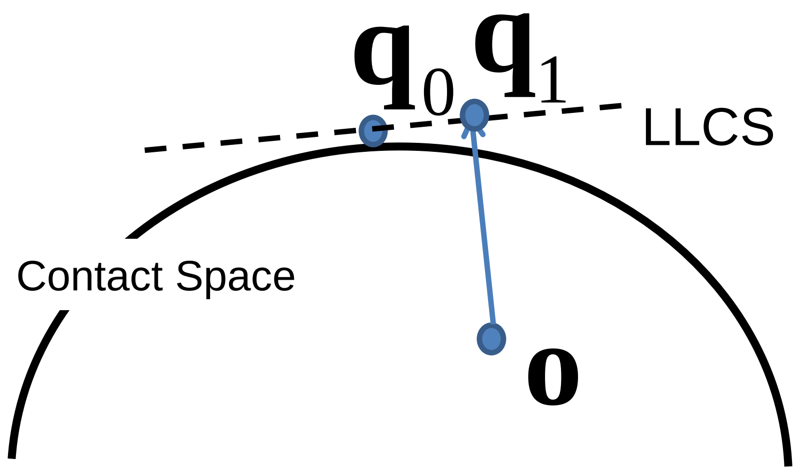
PolyDepth++ Algorithm

2. Contact-space projection



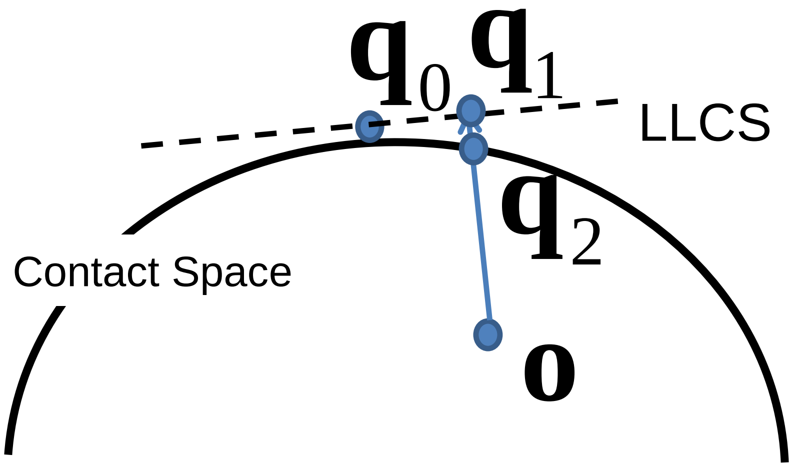
PolyDepth++ Algorithm

3. Constrained optimization



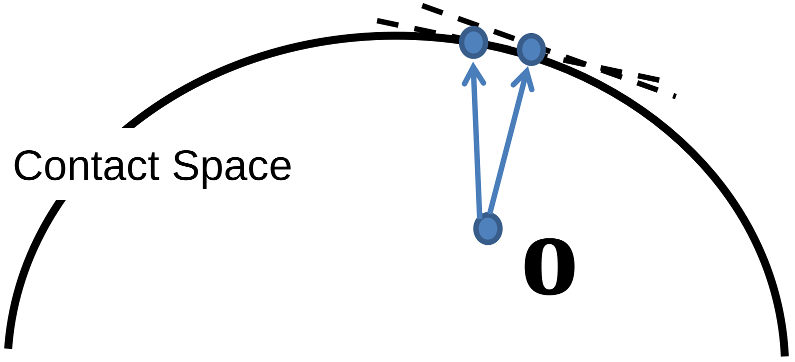
PolyDepth++ Algorithm

4. Re-projection

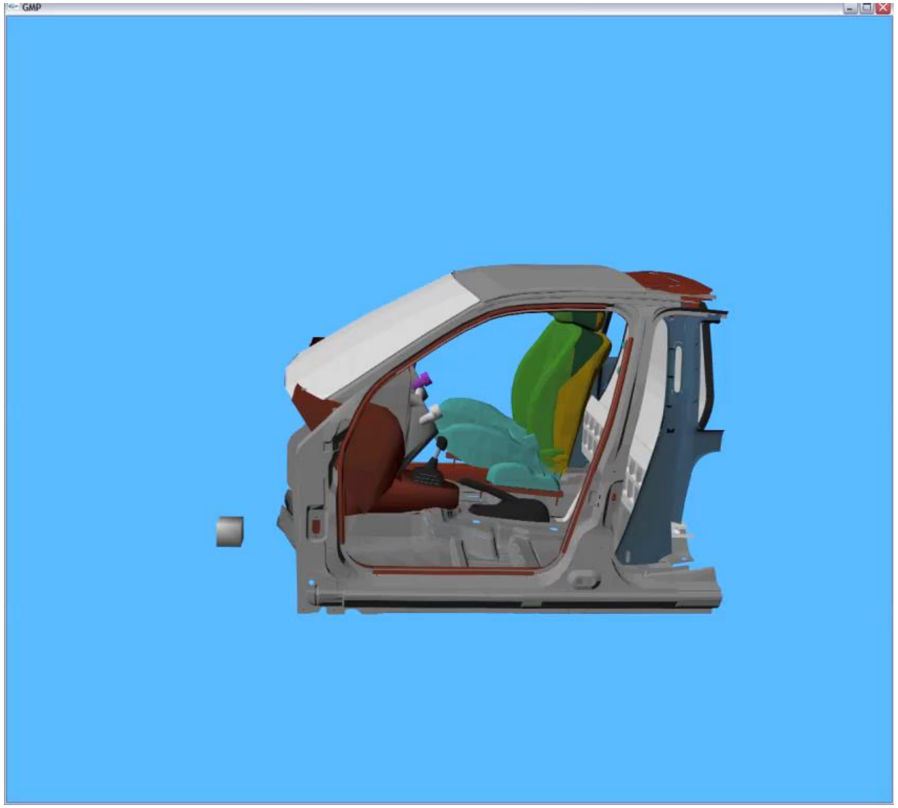


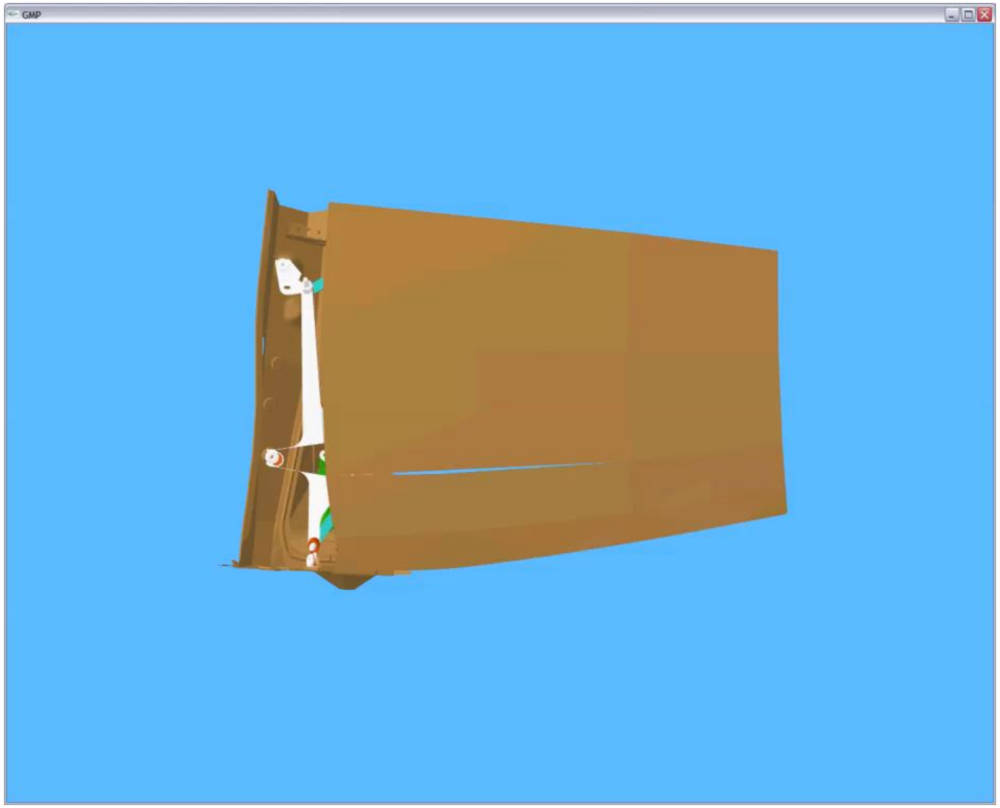
PolyDepth++ Algorithm

5. Iteration until finding a locally-optimal solution



Sampling-based Planner Results





31K triangles (seat), 214K triangles (body)

Planning time: 3 mins

15K triangles (wiper), 12K triangles (body)

Planning time: 20 mins

Optimization-based Planner Results

Non-constraint

Our method

Optimization-based Planner Results

Optimization-based Planner Results

Non-constraint

Our method

Summary

- Continuous collision query
  - Sampling-based motion planning
  - Forward grasp planning

- Motion planning approaches
  - Sampling-based
  - Optimization-based

Acknowledgements

- Min Tang, Youngeun Lee (Ewha)
- Dinesh Manocha (UNC)
- Abderrahmane Kheddar (CNRS/JRL)
- Liangjun Zhang (Samsung)
- Zhixing Xue (FZI/Karlsruhe)
- Kineo Cam (Benchmarking models)

Continuous Collision Detection

- **Hierarchical and Controlled Advancement for Continuous Collision Detection,** *IEEE TVCG 2014*

- **C^2A: Controlled Conservative Advancement for Interactive Continuous Collision Detection**, *IEEE ICRA 2009*

- **Continuous Collision Detection for Non-rigid Contact Computation using Local Advancement**, *IEEE ICRA 2010*

- **Efficient Local Planning using Connection Collision Query,** *WAFR 2010*

Software Implementations

- Source codes are available
  - http://graphics.ewha.ac.kr/FAST (2-manifold)
  - http://graphics.ewha.ac.kr/C2A (polygon soups)
  - http://graphics.ewha.ac.kr/CATCH (articulated)
  - http://wiki.ros.org/fcl (ROS package)

Penetration Depth

- **Interactive Generalized Penetration Depth Computation for Rigid and Articulated Models,** *ACM Transactions on Graphics 2014*
- **Six-degree-of-freedom Haptic Rendering using Translational and Generalized Penetration Depth Computation,** *IEEE World Haptics 2013*
- **PolyDepth: Real-time Penetration Depth Computation using Iterative Contact Space Projection,** *ACM Transactions on Graphics 2012*
- **A Fast and Practical Algorithm for Generalized Penetration Depth Computation,** *Robotics: Science and Systems* , 2007

Software Implementations

- Source codes are available
  - http://graphics.ewha.ac.kr/polydepth (translation, rigid only)
  - http://graphics.ewha.ac.kr/polydepthg (translation and rotation, articulated)

Thank you for listening!

Motion Planning

- Generalized PD is used to retract collision samples to contact samples in sampling-based motion planner.
- A collision-free motion is automatically generated.

http://graphics.ewha.ac.kr