Planning of Graspless Manipulation based on Rapidly-Exploring Random Trees

Kiyokazu MIYAZAWA*   Yusuke MAEDA**   Tamio ARAI***

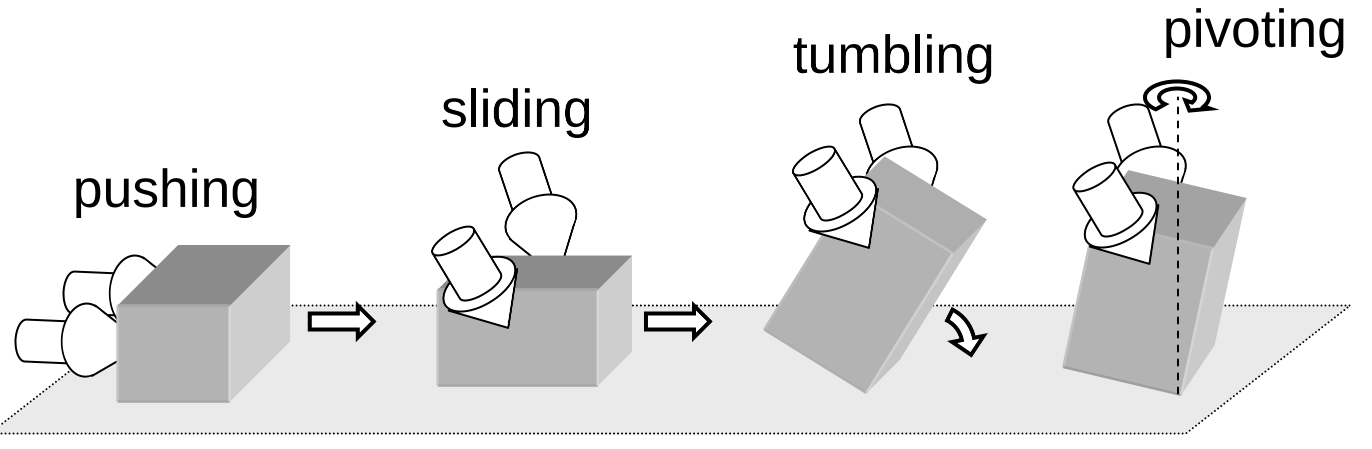
*SONY Corporation

**Yokohama National University   ***The University of Tokyo

1. Introduction

Graspless Manipulation

Manipulation without Grasping [Aiyama 93]



⟺ Manipulation by Grasping (pick-and-place)

Merits of Graspless Manipulation

- <u>Manipulation by smaller force</u>

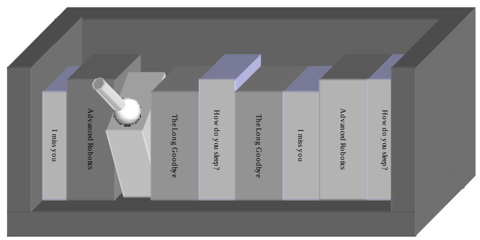
  No need to support all the weight of the object

- <u>Manipulation by simple mechanisms</u>

  Use of environment and gravity as virtual fingers

- <u>Manipulation when grasping is impossible</u>

  e.g. Existence of obstacles



picking up a book from a bookshelf

➡ Graspless manipulation is important as a complement of conventional pick-and-place

Planning of Graspless Manipulation

Problem

Manipulation planning: how to generate robot motion to move an object from initial to goal configuration by graspless manipulation

	Analysis required for Planning	Reversibility of Manipulation
Pick-and-Place	Geometry Level (collision avoidance)	Reversible
Graspless Manipulation	Geometry and Mechanics Level (contact forces and gravity)	Possibly Irreversible

⇨ Planning of graspless manipulation is difficult

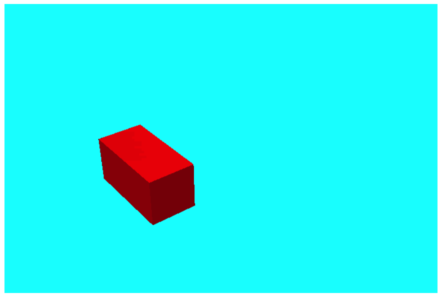
Our Previous Planner

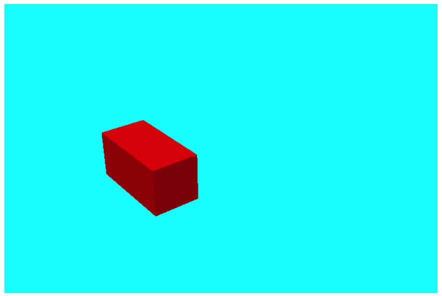
[Maeda ICRA04] [Maeda Adv. Rob. 05]

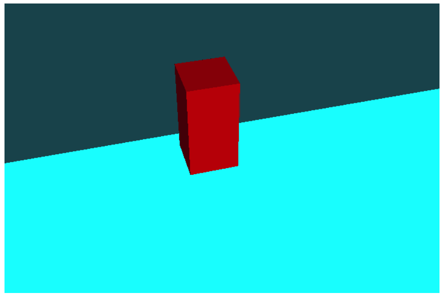
- Planning of graspless manipulation by multiple robot fingertips
- Uniform sampling of C-Space, graph representation of feasible manipulation, and A* search
- Local manipulation feasibility is checked by mechanical analysis [Maeda IROS03]

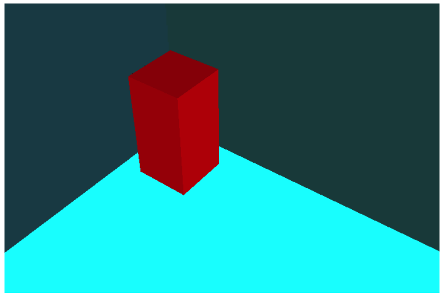
Problem of Our Previous Planner

- Very time-consuming
- Only 1 DOF for manipulated object


533 [CPU min]


203 [CPU min]


84 [CPU min]


990 [CPU min]

(on Pentium 4–2.8GHz)

Objective

Accelerate planning of graspless manipulation

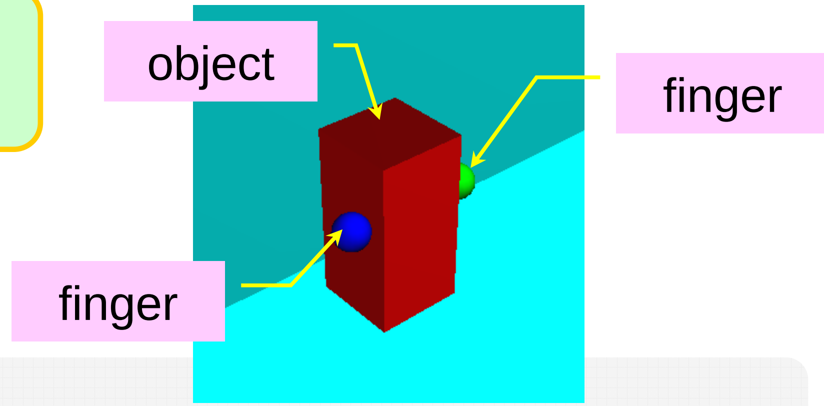
- For various graspless operations by multiple fingers
- With more DOF for manipulated object

Approach

- Give up optimality and find a feasible plan quickly
- Rapidly-exploring Random Trees (RRT)

[LaValle 01]

2. Problem Statement



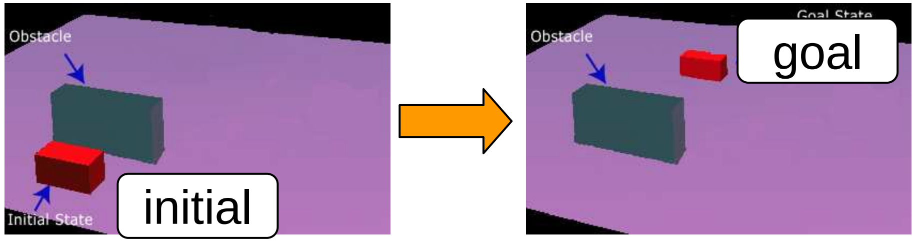
Assumptions

- Quasi-static manipulation of a rigid object
- Under gravity and Coulomb friction
- Each finger is modeled as a sphere
- Finger forces are upper-bounded
- Slipping and rolling of each finger is not allowed
- Each finger is in position- or force-control mode

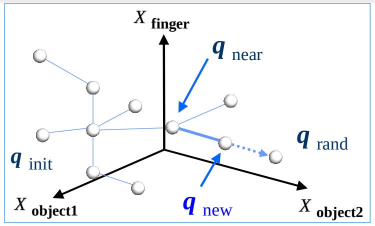
Planning Problem

Input:

• Initial and goal configurations of object

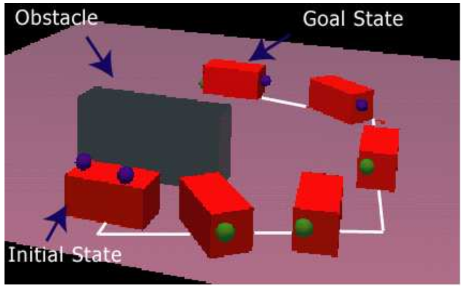


RRT-based representation of feasible manipulation



Output:

• A series of finger control modes and commands



3. Planning of Graspless Manipulation

Configuration of manipulation system

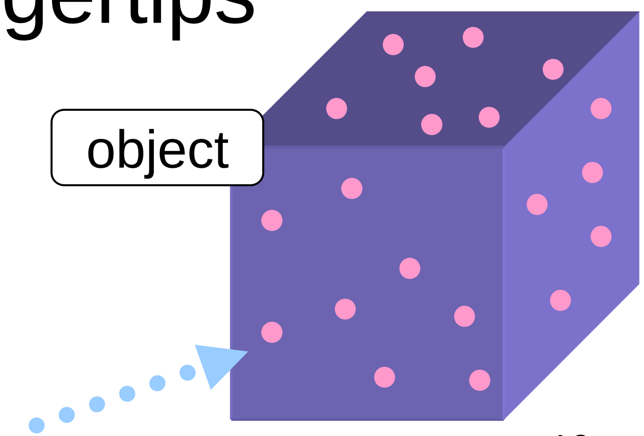
$$\boldsymbol{q} = (\boldsymbol{X}, \boldsymbol{F})$$

- $\boldsymbol{X} = (x, y, \theta)$: Configuration of object
- $\boldsymbol{F} = (\boldsymbol{F}_1, \ldots, \boldsymbol{F}_N)$:

  Configuration of robot fingertips

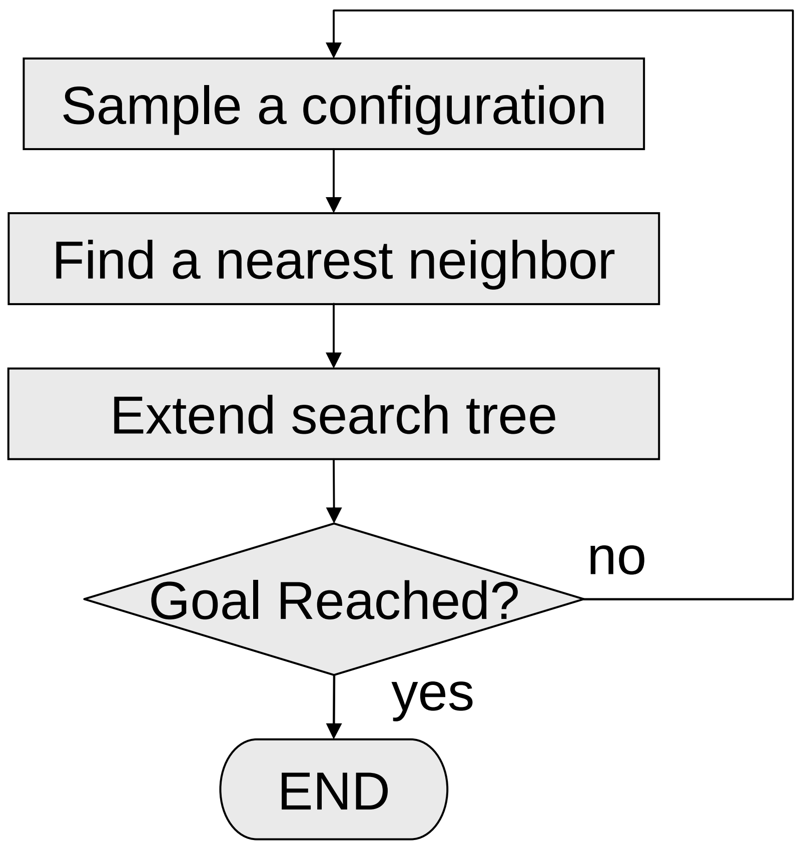
$$\boldsymbol{F}_i = (x_i, y_i, f_i)$$

Face ID



Outline of Planning Algorithm

- Based on RRT-GoalBias [LaValle 01]



Sampling Configurations

- Sample a configuration

$$\boldsymbol{q}_{\text{sample}} = (\boldsymbol{X}_{\text{sample}}, \boldsymbol{F}_{\text{sample}})$$

  – With probability ε, sample the goal configuration (RRT-GoalBias)

  – Otherwise, sample randomly

Finding a Nearest Neighbor

- Find the nearest neighbor in the search tree to the sampled configuration:

$$\boldsymbol{q}_{\text{near}} = (\boldsymbol{X}_{\text{near}}, \boldsymbol{F}_{\text{near}})$$

Distance function:

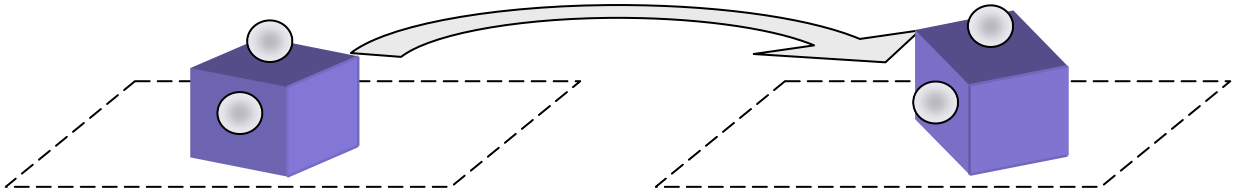
$$d(\boldsymbol{X}_1, \boldsymbol{X}_2) = \sqrt{w_x(x_1 - x_2)^2 + w_y(y_1 - y_2)^2 + w_\theta\left[\text{min}(|\theta_1 - \theta_2|, 2\pi - |\theta_1 - \theta_2|)\right]^2}$$

Just consider the difference between ***object*** configurations

Tree Extension from Nearest Neighbor

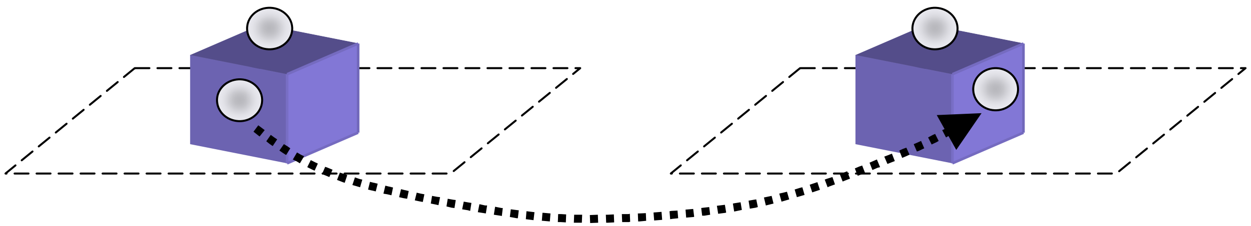
- Determine tree extension target
  - With probability α, change object configuration without changing fingertip locations

$$(\boldsymbol{X}_{\text{near}}, \boldsymbol{F}_{\text{near}}) \rightarrow (\boldsymbol{X}_{\text{sample}}, \boldsymbol{F}_{\text{near}})$$



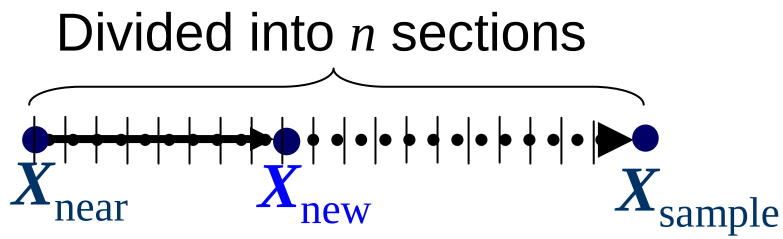
  - Otherwise, change a fingertip location without changing object configuration (Regrasping)

$$(\boldsymbol{X}_{\text{near}}, \boldsymbol{F}_{\text{near}}) \rightarrow (\boldsymbol{X}_{\text{near}}, \boldsymbol{F}_{\text{sample}})$$

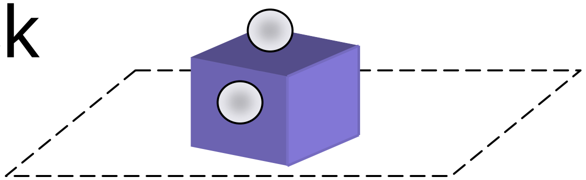


Node Connection

- Changing object configuration
  - Collision check
  - Manipulation feasibility check [Maeda IROS03]



- Changing fingertip location
  - Collision check
  - Stability check

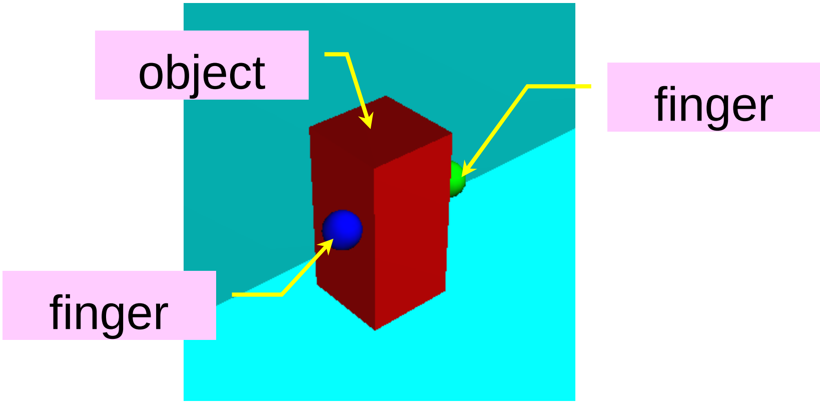


Fingertip Locations

- Restrict fingertip locations in a set of candidate points
  - Based on Halton sequence
  - When node-connection error rate grows, new candidate points are added

4. Planned Results

Graspless Manipulation of a Cuboid by Two Robot Fingers



Mass of object = 1

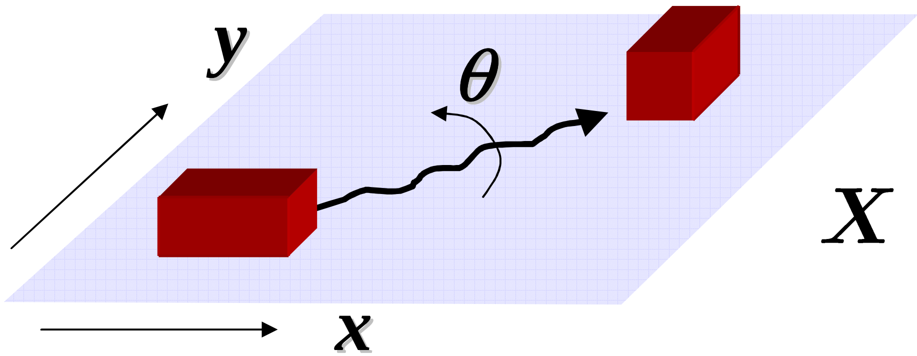
Friction coef. between environment and object = 0.5

Friction coef. between fingers and object = 0.7

Maximum finger forces = 10

Acceleration of gravity = 9.8

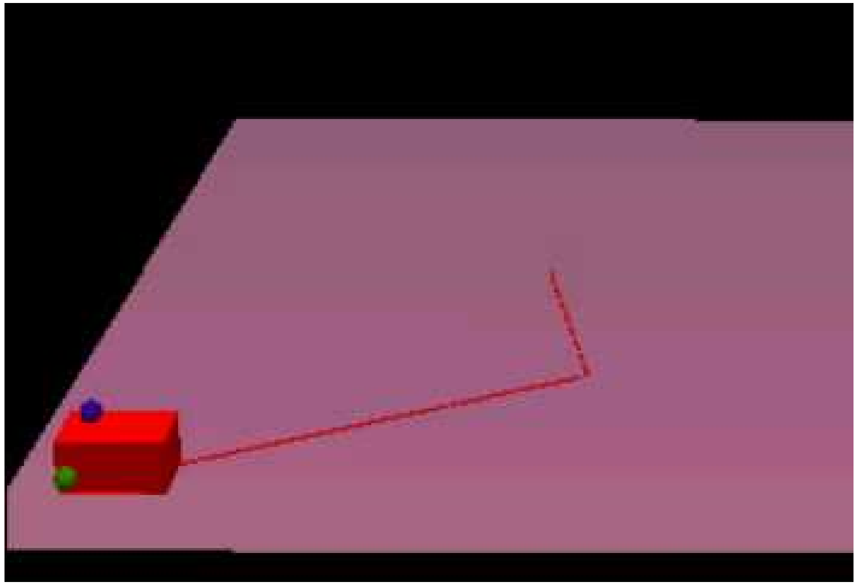
Plan A: Sliding on a Plane



$$\boldsymbol{X} = (x, y, \theta)$$

Start		Goal
(10, 10, 0)	→	(50, 50, π/2)

Plan A: Result



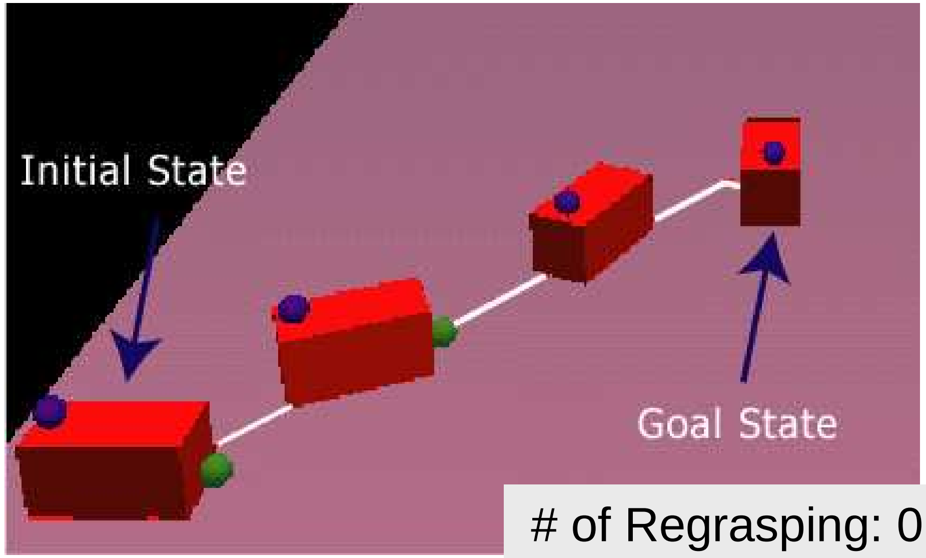
- fingertip in force-control mode (blue)
- fingertip in position-control mode (green)

Planning Time: 4.4 [CPU min]
(on Pentium4–2.8GHz)

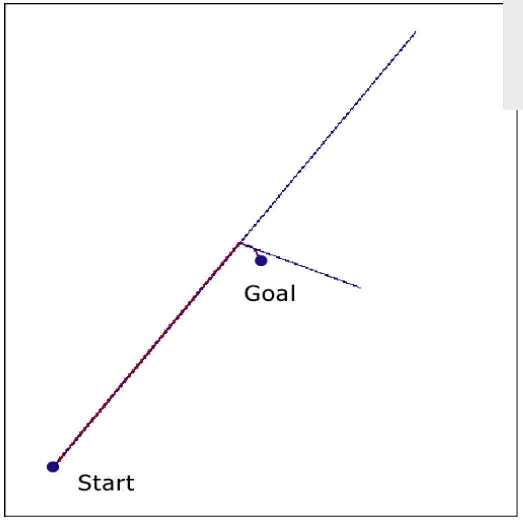
Planning Time Range [CPU min]	**0.2 ~ 78.8**
Avg. Planning Time [CPU min]	**13.0**
Avg. # of Regrasp	**1.5**
Avg. # of Sampling	**28.0**

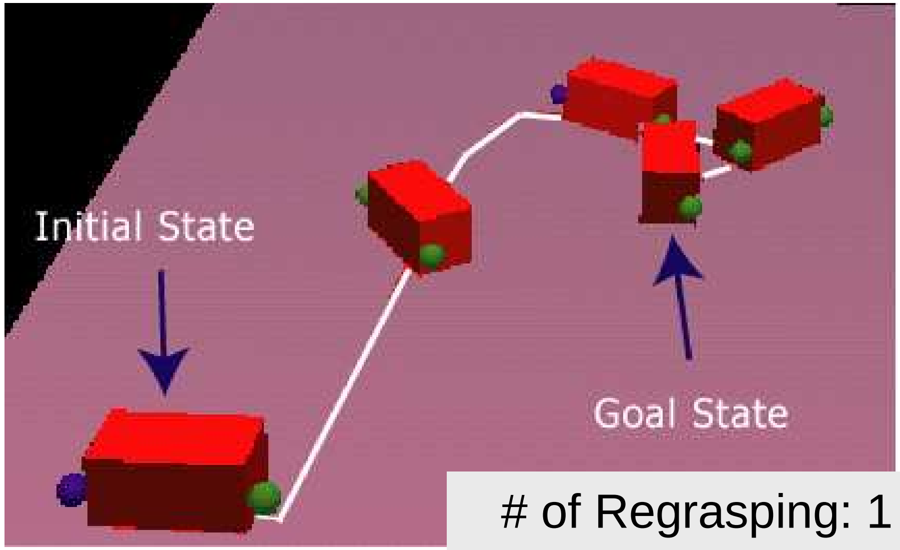
(in 100 trials)

Plan A: Other Results

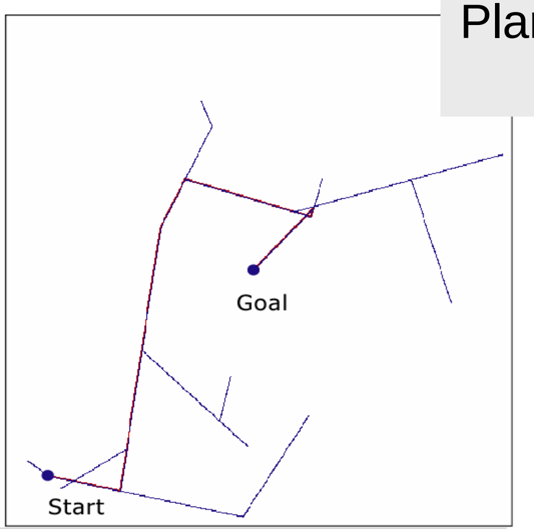


# of Regrasping: 0
Planning Time: 4.1 [CPU min]

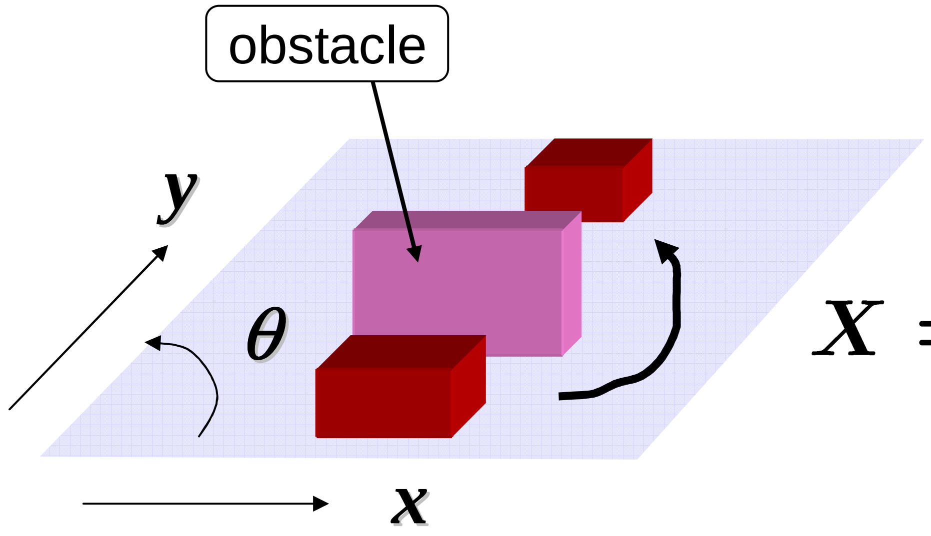




# of Regrasping: 1
Planning Time: 18.7 [CPU min]



Plan B: Sliding on a Plane

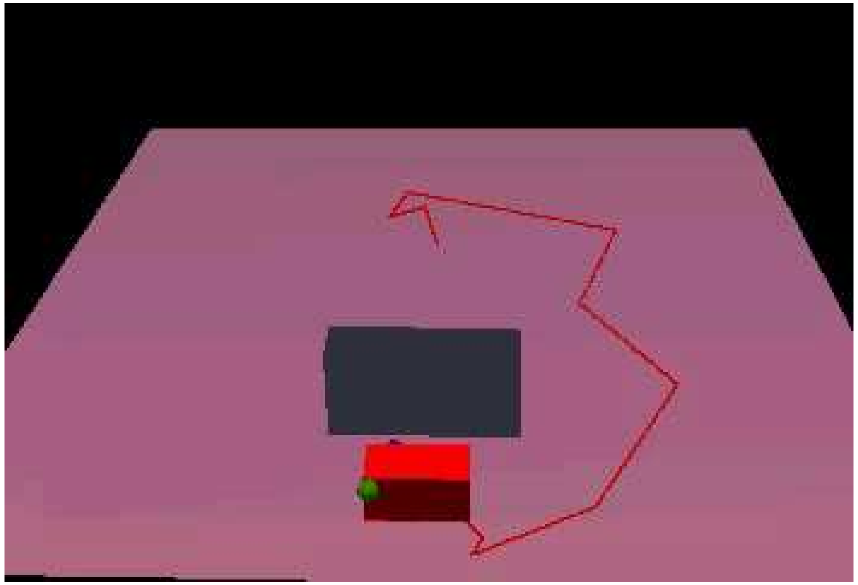


$$X = (x, y, \theta)$$

Start → Goal

(50, 10, 0) → (50, 60, 0)

Plan B: Result



● fingertip in force-control mode

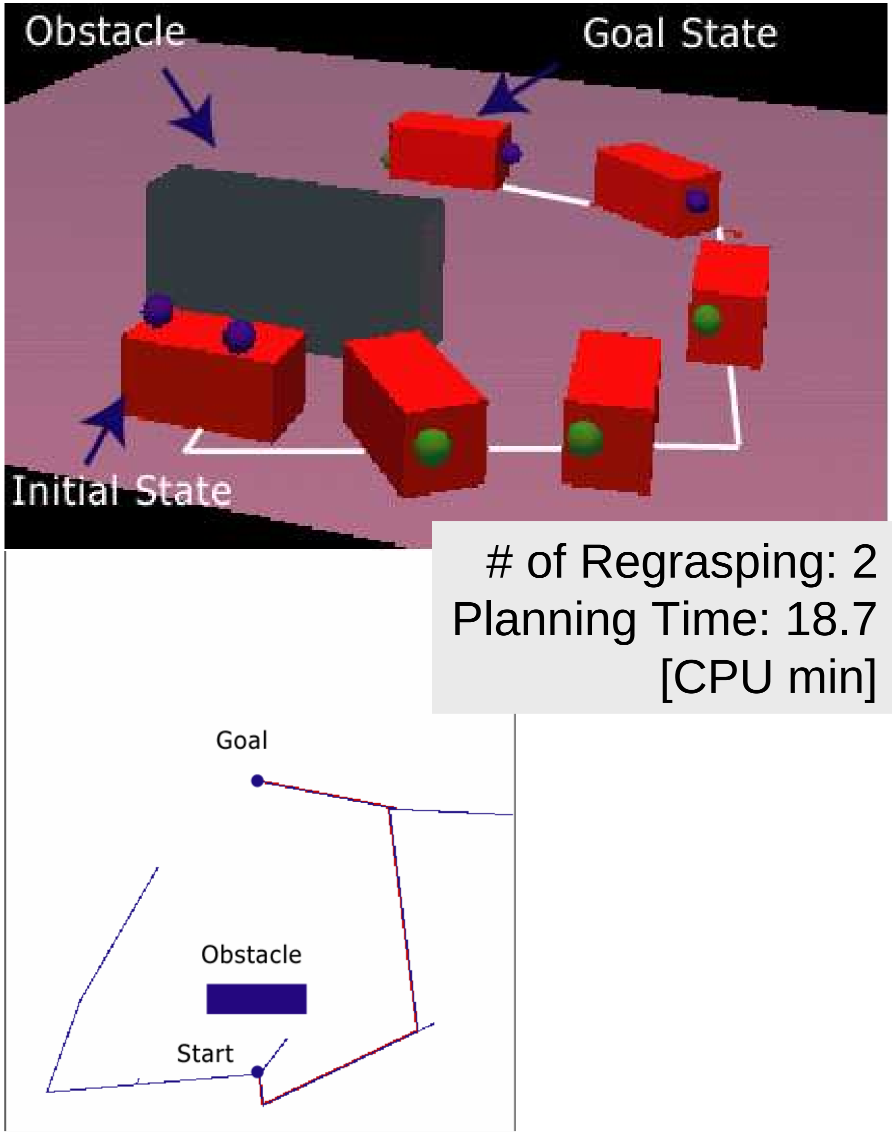
● fingertip in position-control mode

Planning Time: 29.8 [CPU min]
(on Pentium4–2.8GHz)

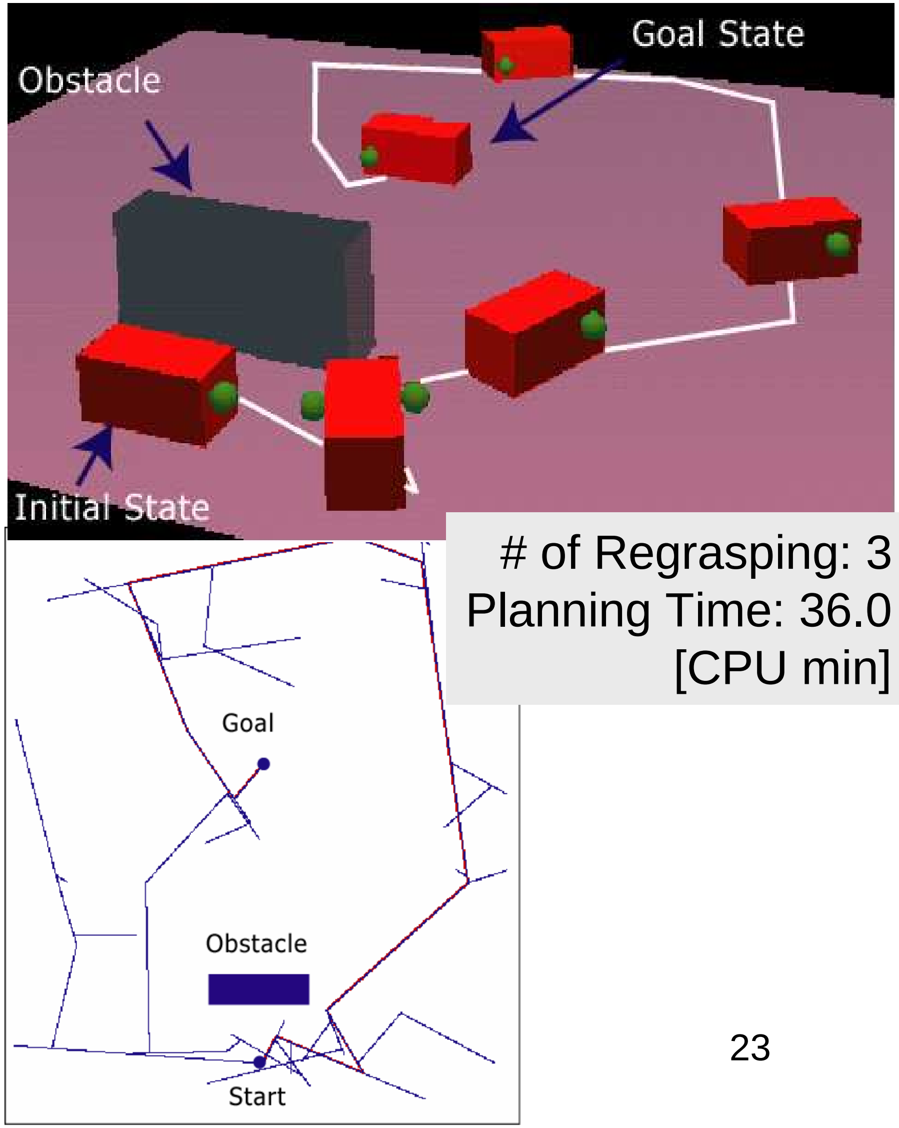
Planning Time Range [CPU min]	**0.9 ~ 57.0**
Avg. Planning Time [CPU min]	**15.1**
Avg. # of Regrasp	**2.3**
Avg. # of Sampling	**47.1**

(in 100 trials)

Plan B: Other Results

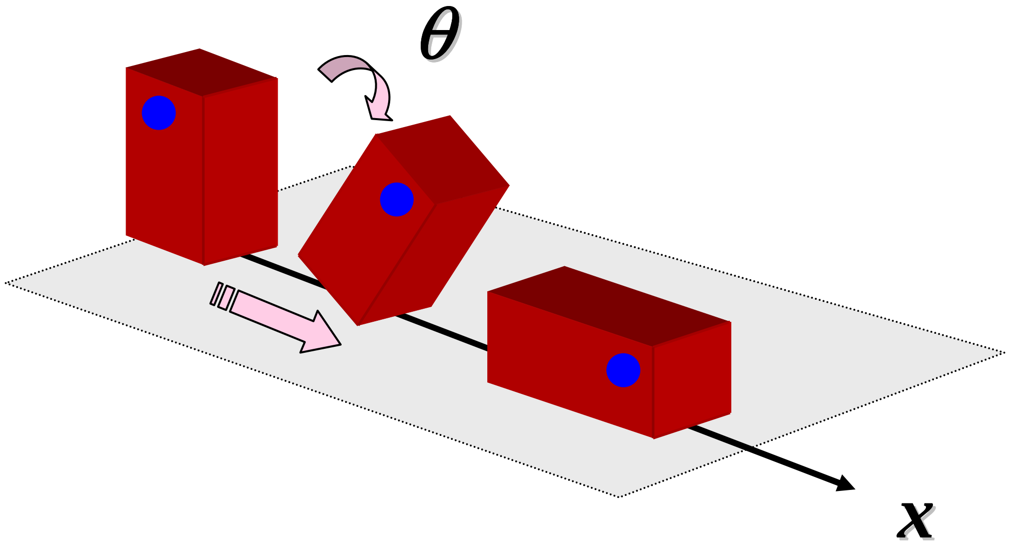


# of Regrasping: 2
Planning Time: 18.7 [CPU min]



# of Regrasping: 3
Planning Time: 36.0 [CPU min]

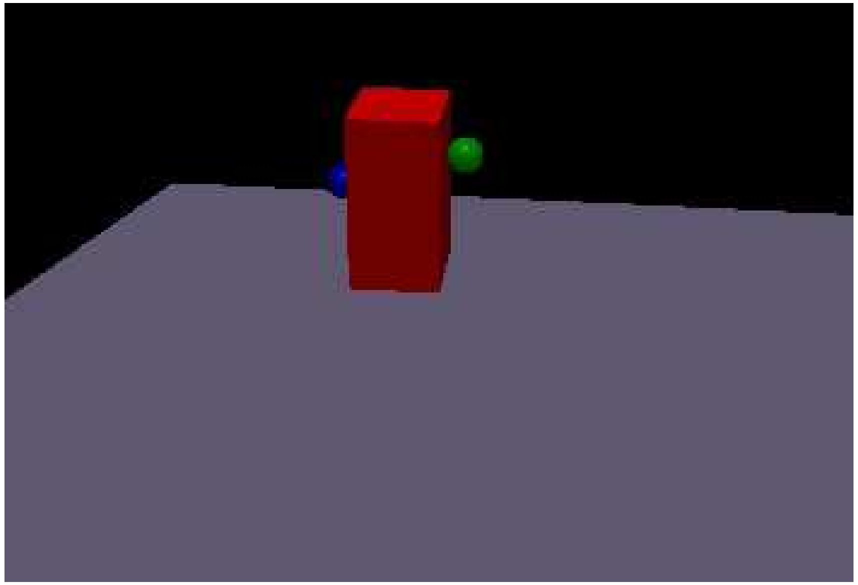
Plan C: Tumbling and Sliding on a Plane



$$\boldsymbol{X} = (x, \theta)$$

Start		Goal
(0, 0)	→	(20, π/2)

Plan C: Results



- fingertip in force-control mode (blue)
- fingertip in position-control mode (green)

Planning Time: 3.7 [CPU min]
(on Pentium4–2.8GHz)

Planning Time Range [CPU min]	**0.9 ~ 37.4**
Avg. Planning Time [CPU min]	**5.8**
Avg. # of Regrasp	**3.1**
Avg. # of Sampling	**513**

(in 100 trials)

5. Conclusion

A planner of graspless manipulation is developed.

- Manipulation planning where the object has two or three DOF
- Find a feasible manipulation quickly based on RRT

Future Work

- Incorporation of appropriate heuristics