Motion Planning of Robot Fingertips for Graspless Manipulation

Yusuke MAEDA* Tomohisa NAKAMURA** Tamio ARAI***

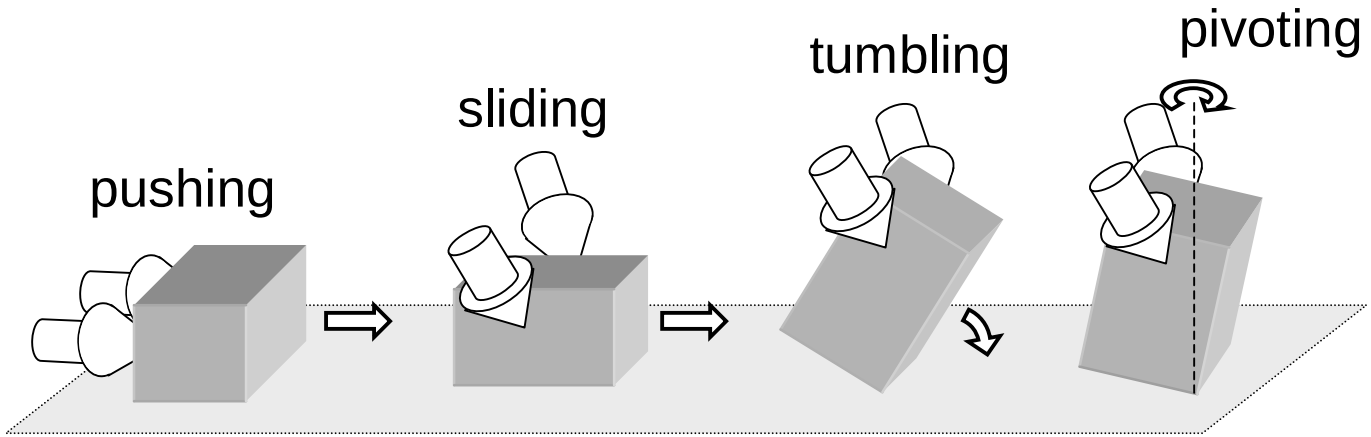
*Yokohama National University
**NTT Data Corporation ***The University of Tokyo

1. Introduction
2. Problem Statement
3. Planning of Graspless Manipulation
4. Planned Results and Experiments
5. Conclusion

1. Introduction

Graspless Manipulation

Manipulation without Grasping [Aiyama 1993]
(Nonprehensile Manipulation)



⟺ Manipulation by Grasping
(pick-and-place)

Merits of Graspless Manipulation

- Manipulation by smaller force

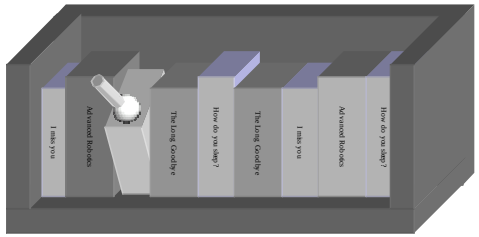
  No need to support all the weight of the object

- Manipulation by simple mechanisms

  Use of environment and gravity as virtual fingers

- Manipulation when grasping is impossible

  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 manipulate 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

Planning of Graspless Manipulation

Related Works

- Motion Planning of Manipulated Object

  [Yu 96] [Marigo 00] [Ji 01] [Aiyama 01]…

- Planning of Manipulation by Specific Operation

  (Pushing) [Kurisu 94] [Lynch 96]…

  (Tumbling) [Sawasaki 89] [Yamashita 03]…

  (Other) [Trinkle 93] [Terasaki 98] [Erdmann 98]…

Objective

Planning of General Graspless Manipulation

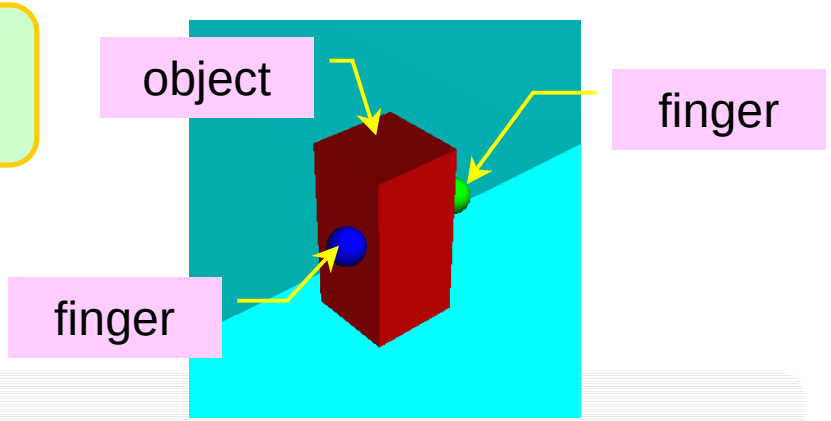
- For various graspless operations
- For manipulation by multiple fingers

Approach

Extension of our previous method [Maeda 2001ICRA]

- Directed graph representation of feasible manipulations
- Graph searching with A* for manipulation planning
- Generation of robust manipulation considering the stability of manipulation

2. Problem Statement



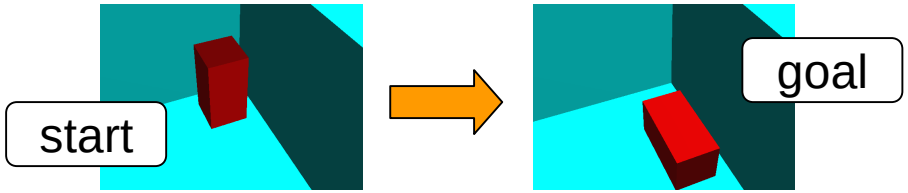
Assumptions

- 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

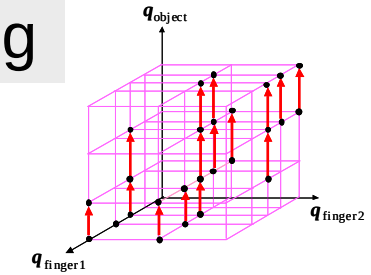
Planning Problem

Input:

- Initial and goal configurations of object

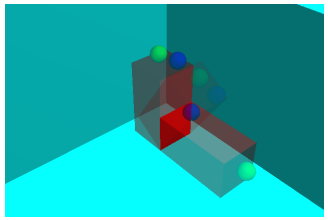


Graph construction and searching



Output:

- A series of finger control modes and commands



3. Planning of Graspless Manipulation

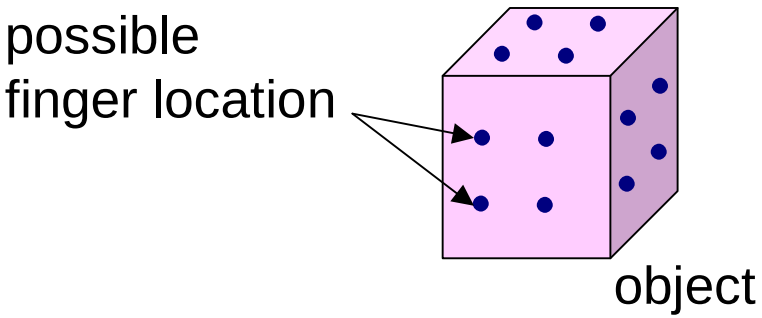
Graph Representation of Feasible Manipulation

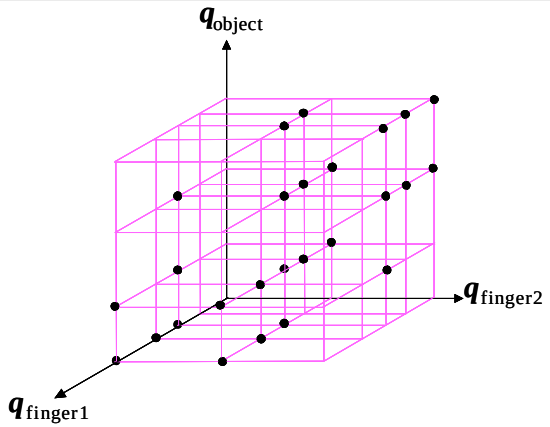
- Consider $(M + 2N)$-dimensional C-Space

  (M : d.o.f of object, N : # of fingers)

  ■ possible finger locations are restricted on the object surface

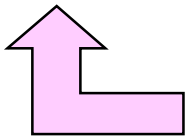
- We make nodes by discretizing C-Space



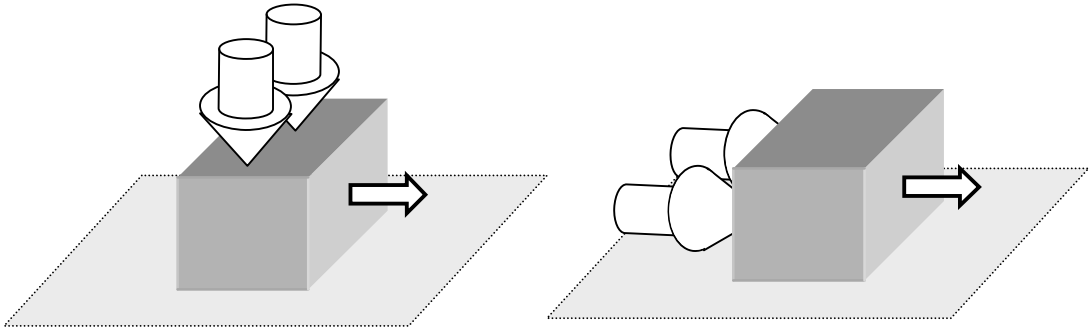


Arc Generation

- Nodes are connected by directed arcs when <u>manipulation is feasible with sufficient stability</u>



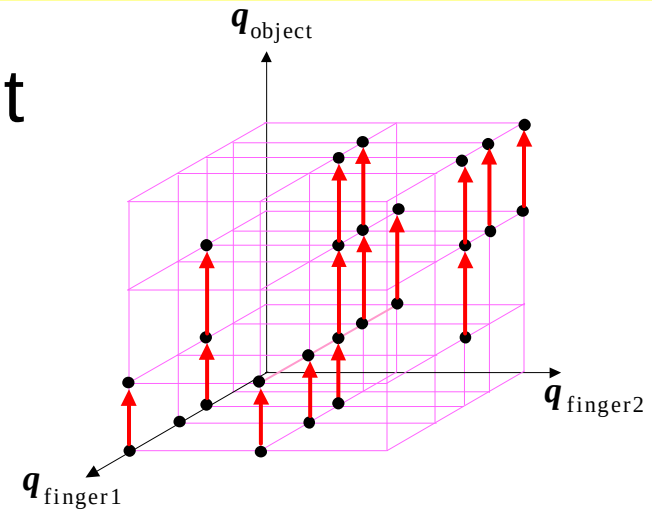
Feasibility can be judged
by finding a combination of finger control modes and commands that maximizes manipulation stability [Maeda 2003IROS]

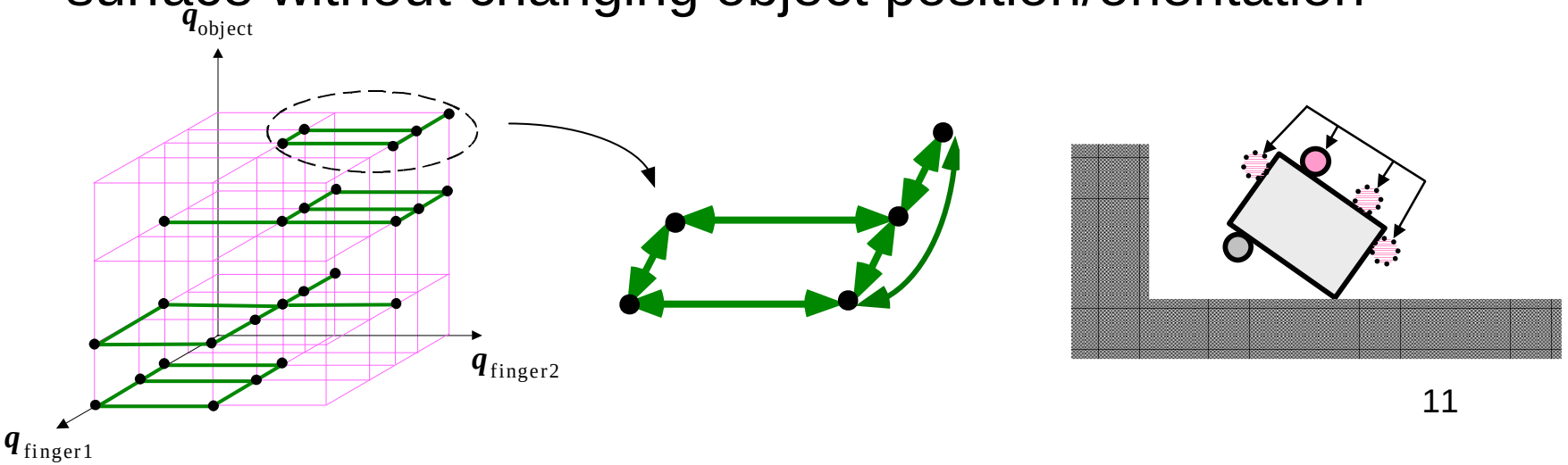


Force Control or Position Control? [Maeda 2003IROS]

Two Kinds of Arcs

- Arcs for object displacement
  - Manipulating object without changing fingertip locations on object surface
- Arcs for regrasping
  - Reposition of one robot fingertip on object surface without changing object position/orientation





Manipulation Planning by Graph Searching

Constraints

- Avoid unstable manipulation

  ⇨ Only arcs with large stability is adopted $(z \geq z_{\mathrm{min}})$

Cost Assignment

- Minimize the number of regrasping
- Minimize the displacement of fingertips
- Maximize the manipulation stability

$$C = \begin{cases} \max\limits_{i} \sum\limits_{j=1}^{P} \left(1 + \dfrac{X_{\mathrm{stab}}}{z_j}\right) \|\Delta \boldsymbol{q}_{\mathrm{finger}\,i,j}\| & \text{(cost for object displacement)} \\ X_{\mathrm{regr}} \quad \text{(cost for regrasping)} & (X_{\mathrm{regr}} \gg 1 \gg X_{\mathrm{stab}}/z_{\mathrm{min}}) \end{cases}$$

Planning by A* Search

Heuristic Function for A*

$$H = \begin{cases} \max_i \|\Delta \boldsymbol{q}^*_{\text{finger}\,i}\| \\ n_{\text{viol}} X_{\text{regr}} \end{cases}$$

If current fingertip locations are geometrically feasible even in goal configuration

If current fingertip locations are geometrically infeasible in goal configuration

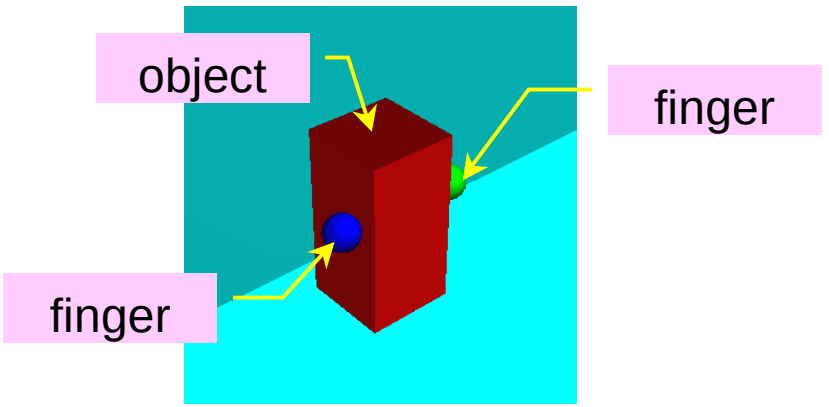
$\|\Delta \boldsymbol{q}^*_{\text{finger}\,i}\|$: estimated displacement of i-th finger to goal

n_{viol} : # of fingertips whose locations are infeasible in goal

➡ Find optimal solution by admissible heuristic function

4. Planned Results and Experiments

Example: Graspless Manipulation of a Cuboid by Two Robot Fingers



Mass of object = 1

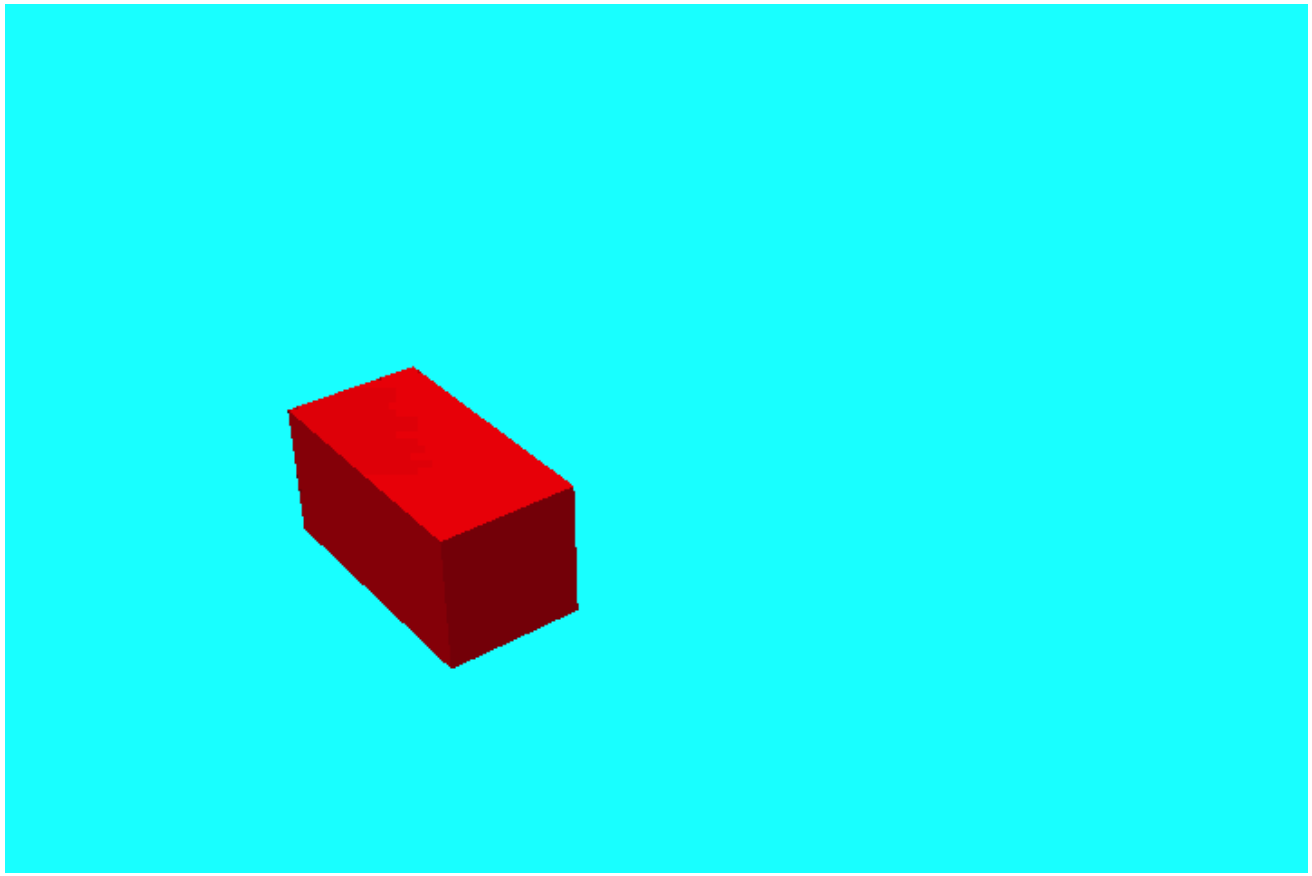
Friction coef. between environment and object = 0.5

Friction coef. between fingers and object = 0.7

Maximum finger forces = 10 (or 5)

Acceleration of gravity = 9.8

Planned Sliding (for “Strong” Fingers)

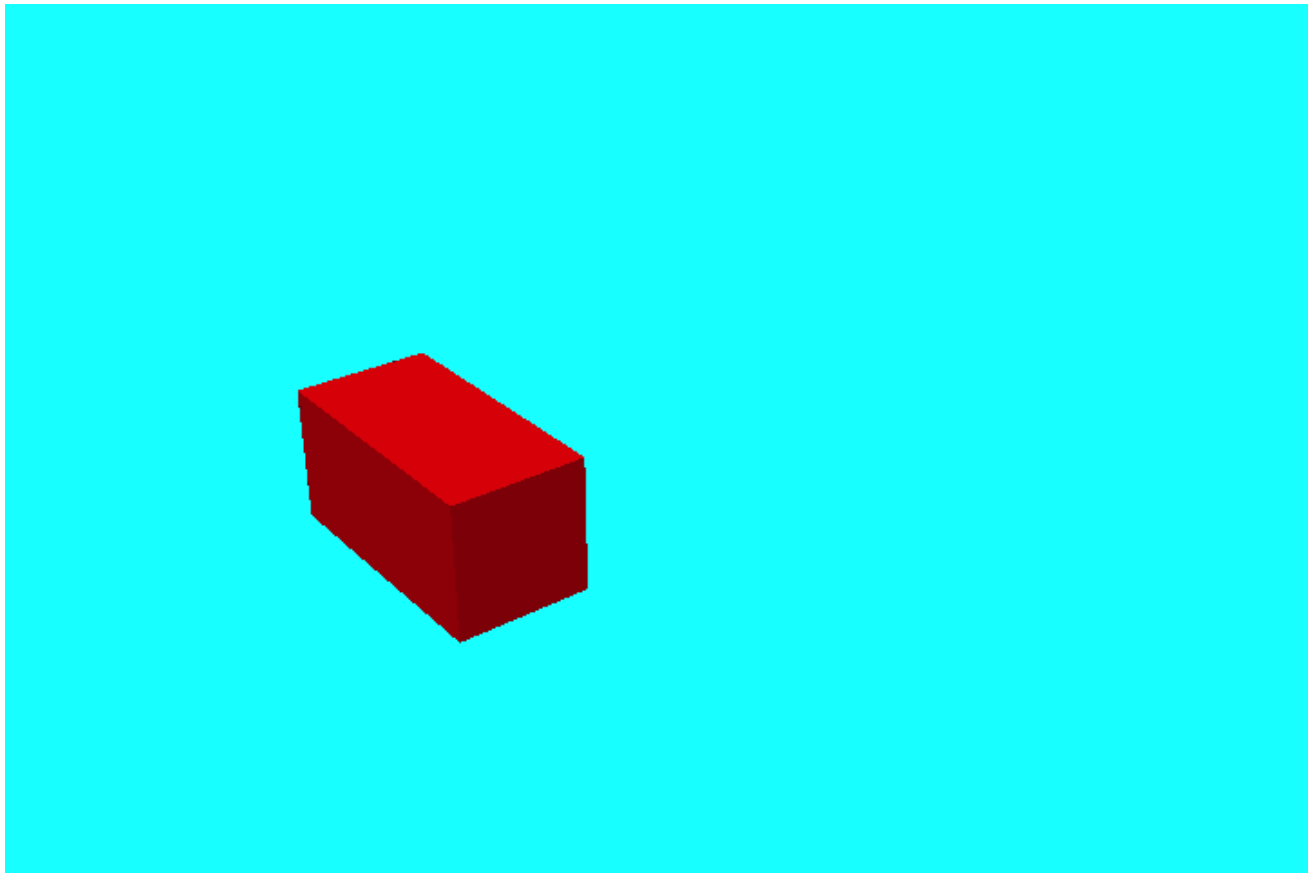


Force Control Mode Position Control Mode

Planning Time: 533 CPU Min. (Pentium4–2.8GHz)

# of expanded nodes: 32 538 / 1 335 201

Planned Sliding (for “Weak” Fingers)

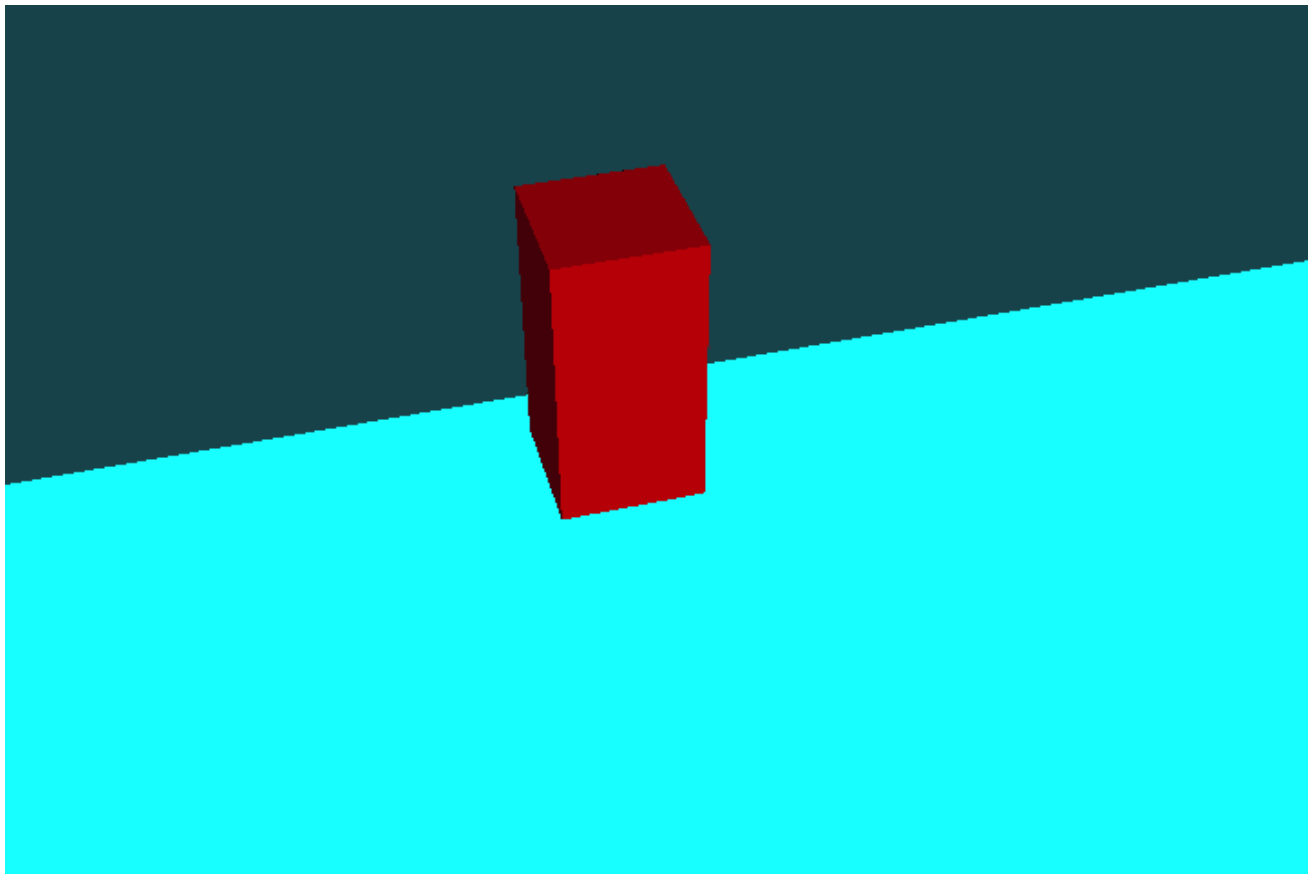


Force Control Mode Position Control Mode

Planning Time: 203 CPU Min. (Pentium4–2.8GHz)

# of expanded nodes: 23 915 / 1 335 201

Planned Tumbling (with a Wall)

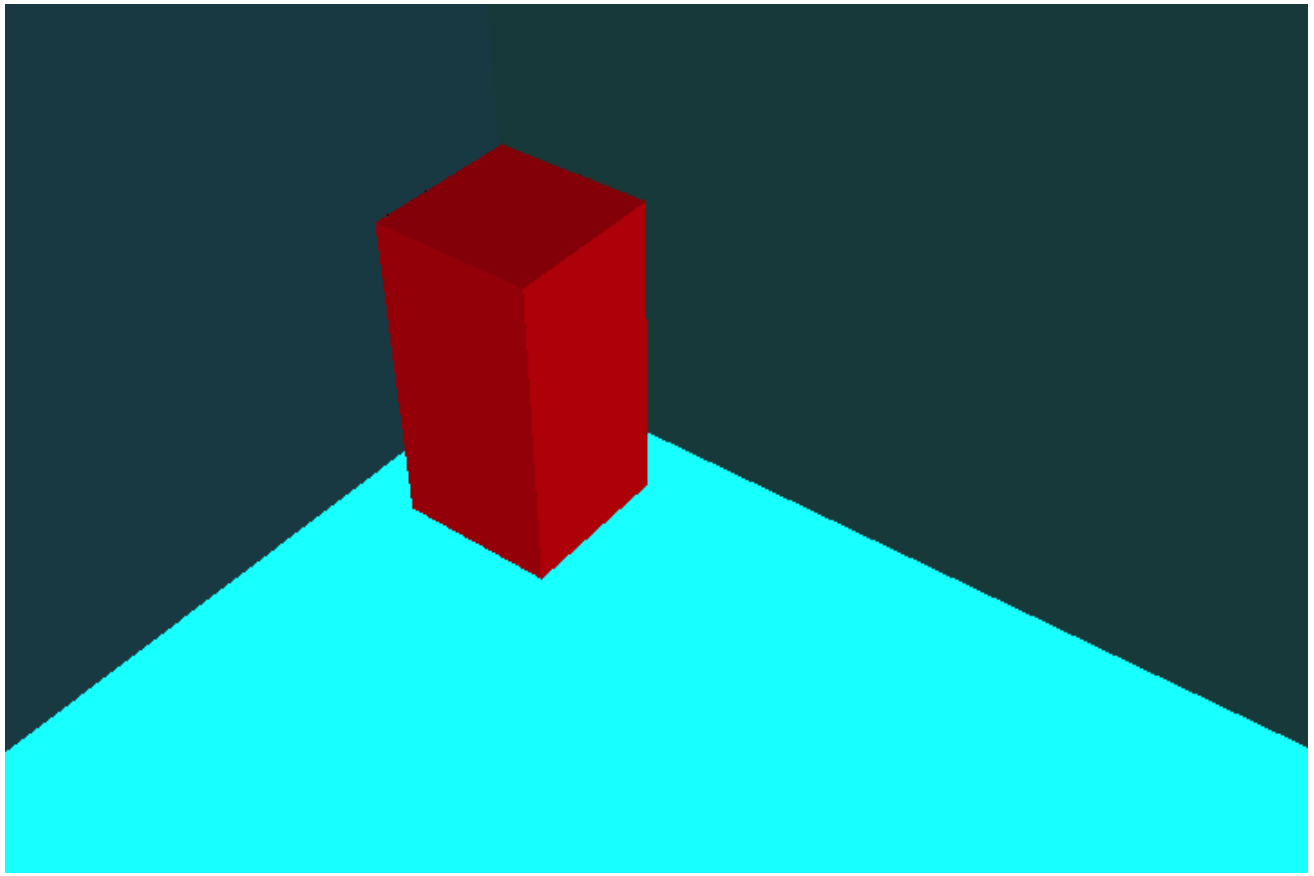


Force Control Mode Position Control Mode

Planning Time: 84 CPU Min. (Pentium4–2.8GHz)

# of expanded nodes: 1 923 / 1 335 201

Planned Tumbling (with Two Walls)



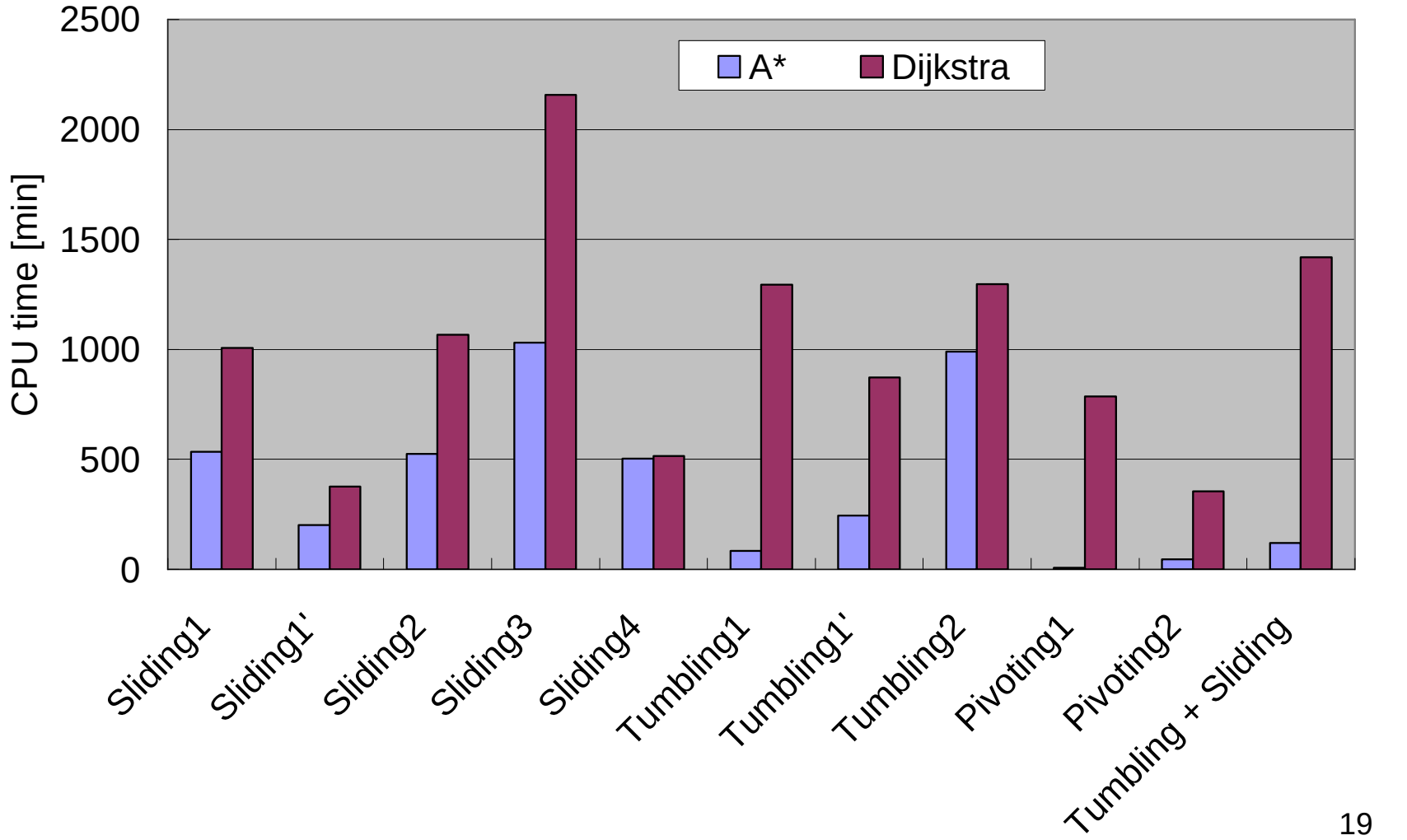
Force Control Mode Position Control Mode

Planning Time: 990 CPU Min. (Pentium4–2.8GHz)

# of expanded nodes: 74 028 / 1 335 201

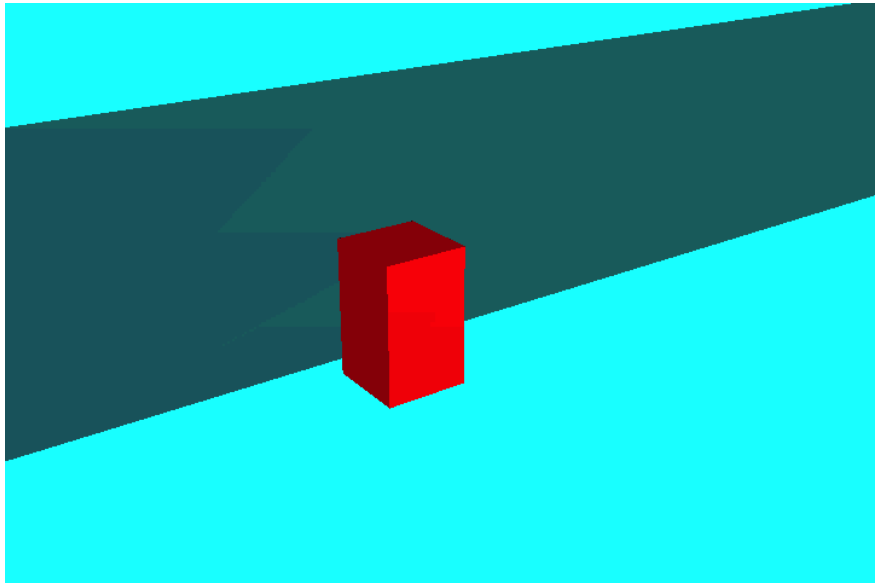
Computation Time for Planning

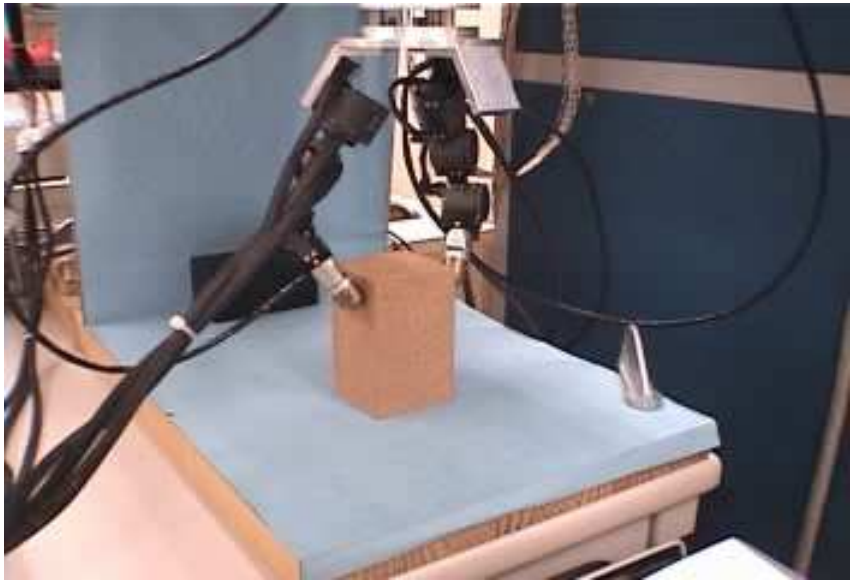
(Pentium4–2.8GHz)



Execution of Planned Manipulation

Tumbling by two robot fingers





5. Conclusion

A planning algorithm of graspless manipulation is developed.

◆ Various graspless operations can be generated: pushing, tumbling, etc.

◆ An example of execution of planned manipulation by an actual robot is shown.

Future Work

- Reduction of computation
  - ➢ Randomized Motion Planning Techniques