Multi-Rate Multi-Range Dynamic Simulation for Haptic Interaction

Ikumi Susa,
Makoto Sato,
Shoichi Hasegawa
Tokyo Institute of Technology

Multi-Rate System for Haptic Interaction

- Motivation
  - Complicated virtual world
    - high polygon, multi-body, deformable objects, fluid
  - High quality haptic display
    - stable, stiff, friction, texture

・Intermediate Representation

[Adachi *et al*., 1995], [Mark *et al*., 1996], [Hasegawa *et al*., 1999]

・Virtual Coupling

[Otaduy *et al*., 2006], [Akahane *et al*., 2006]

・Constraint based Coupling

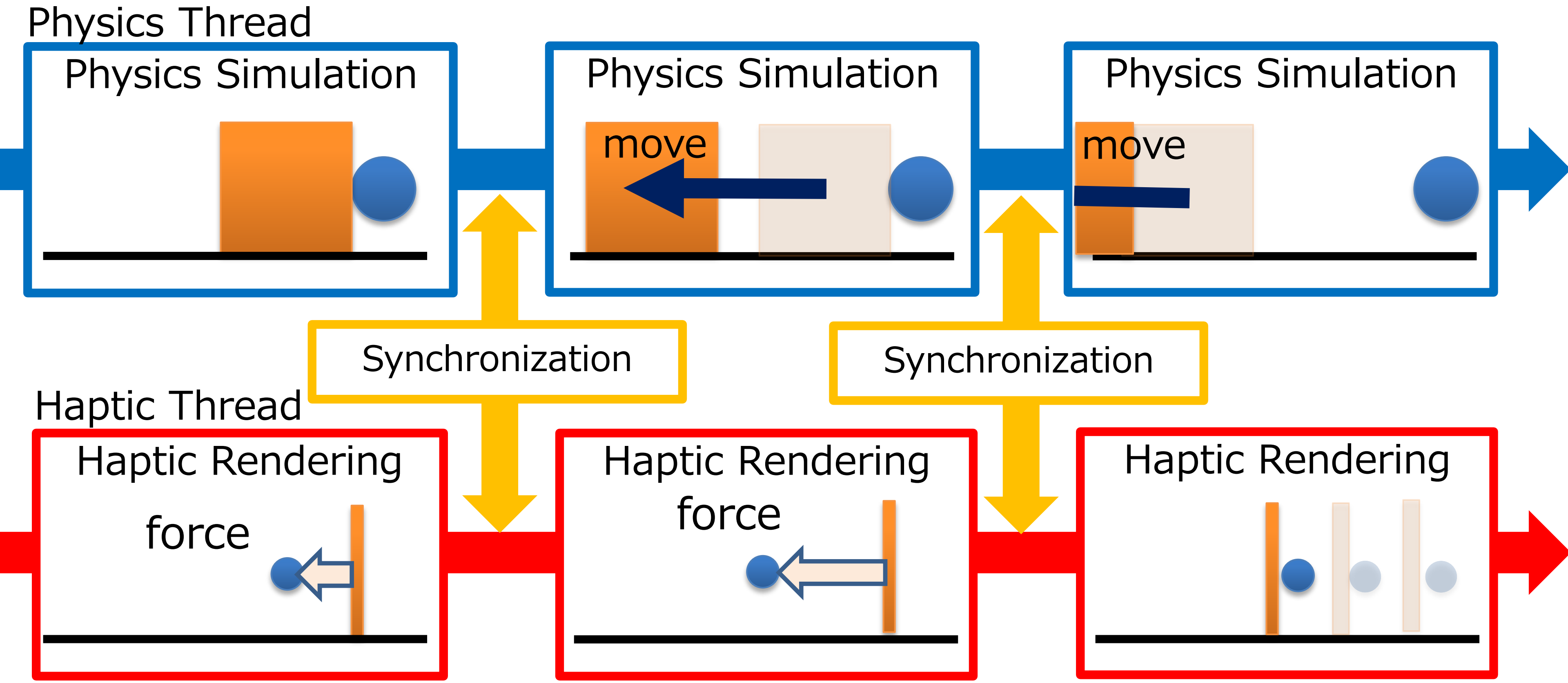
[Ortega *et al*., 2007]

・Deformable Object

[Mendoza and Laugier, 2001]

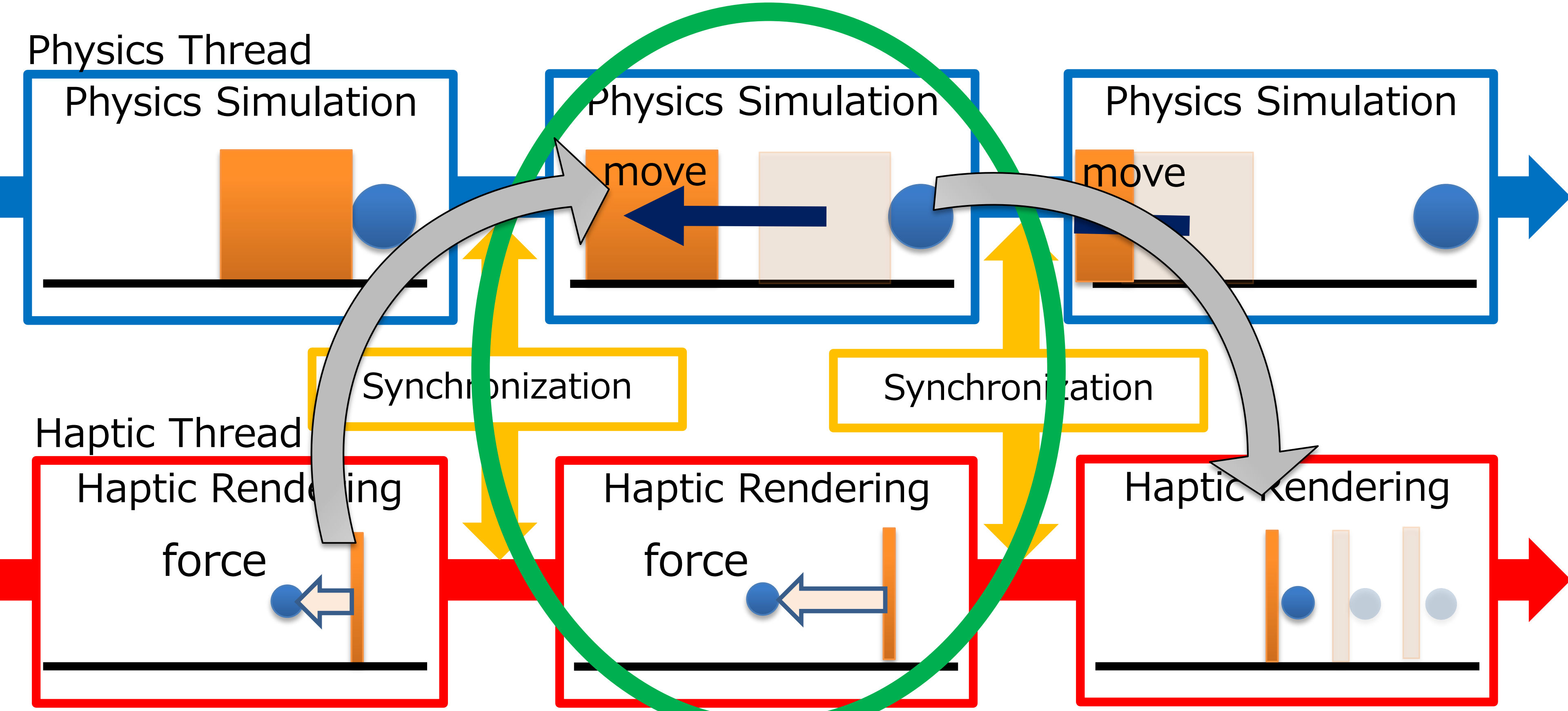
Haptic Artifacts

- Multi-thread synchronization causes time delay
- Time delay makes discrepancy between physics simulation and haptic rendering



Haptic Artifacts

- Multi-thread synchronization causes time delay
- Time delay makes discrepancy between physics simulation and haptic rendering



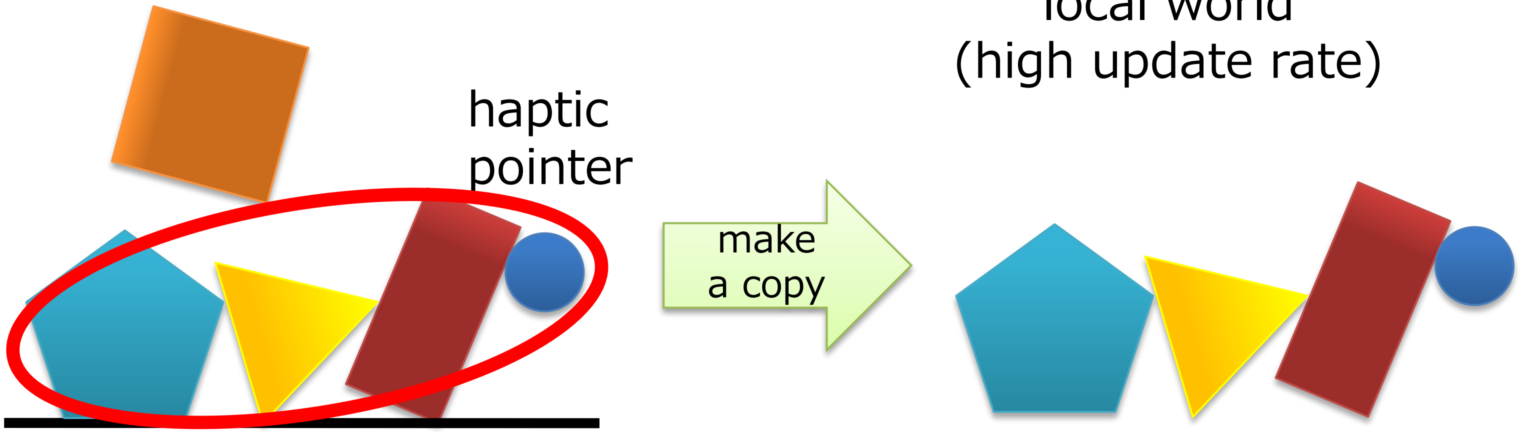
Related Work

[Glondu *et al.* 2010]
A new coupling scheme for haptic rendering of rigid bodies interactions based on a haptic sub-world using a contact graph

- Building a local world in relation to a global world
- Making a contact graph

global world
(low update rate)

local world
(high update rate)



haptic
pointer

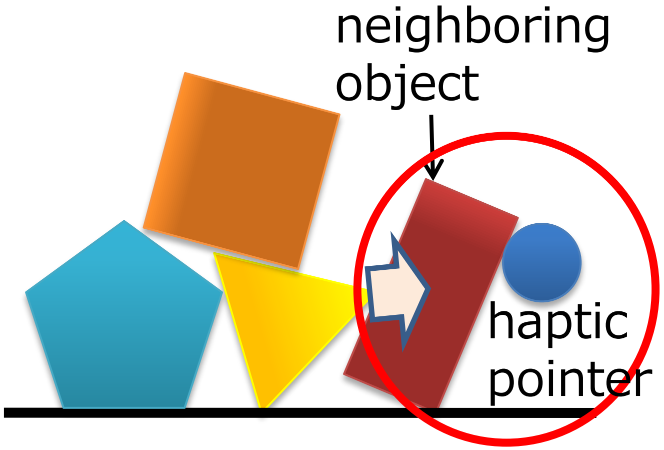
make
a copy

A group of objects
in contact with each other

- haptic rendering
- local dynamics simulation

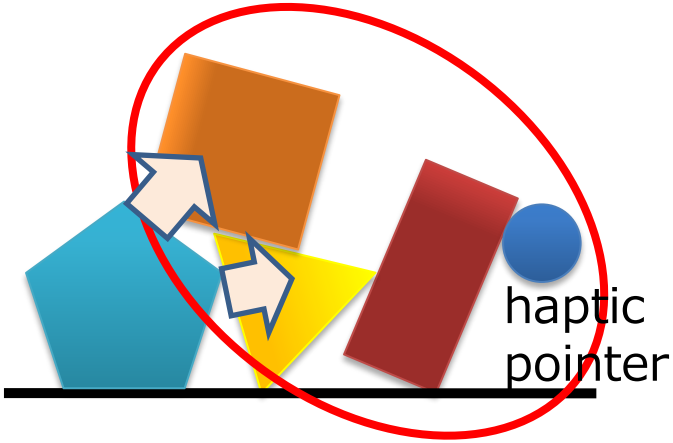
Range of the Local World

Neighboring objects
of the haptic pointer



Missing the contact forces

Nth object
from the haptic pointer



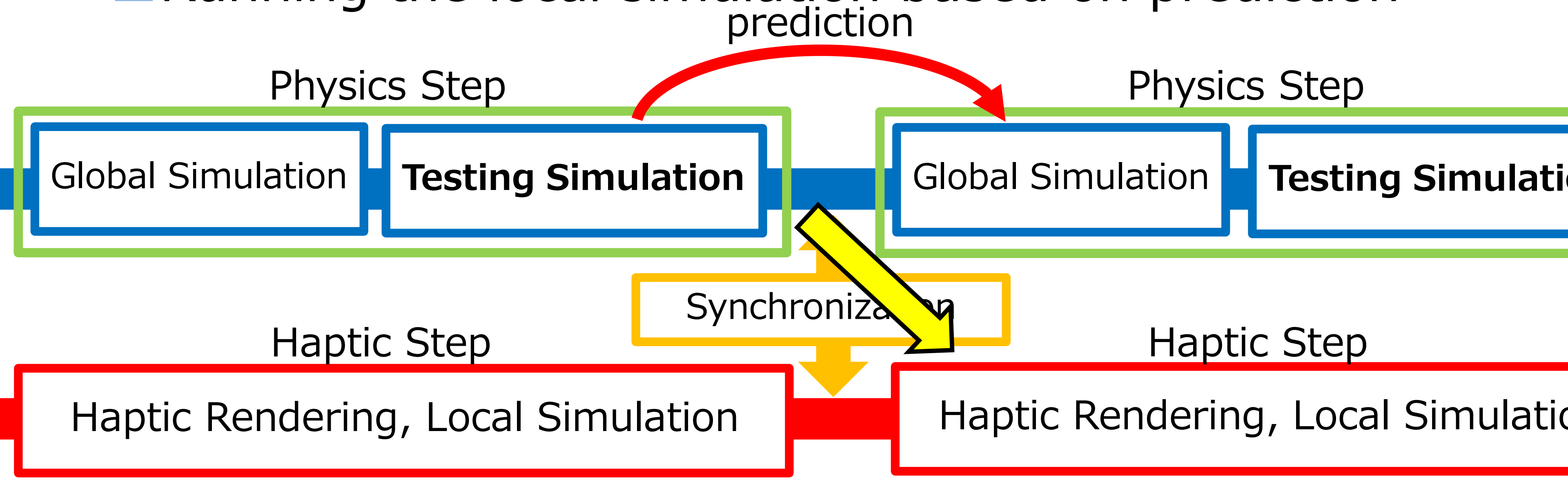
Same problem will cause

Difficult to display adequate force feedback

Approach

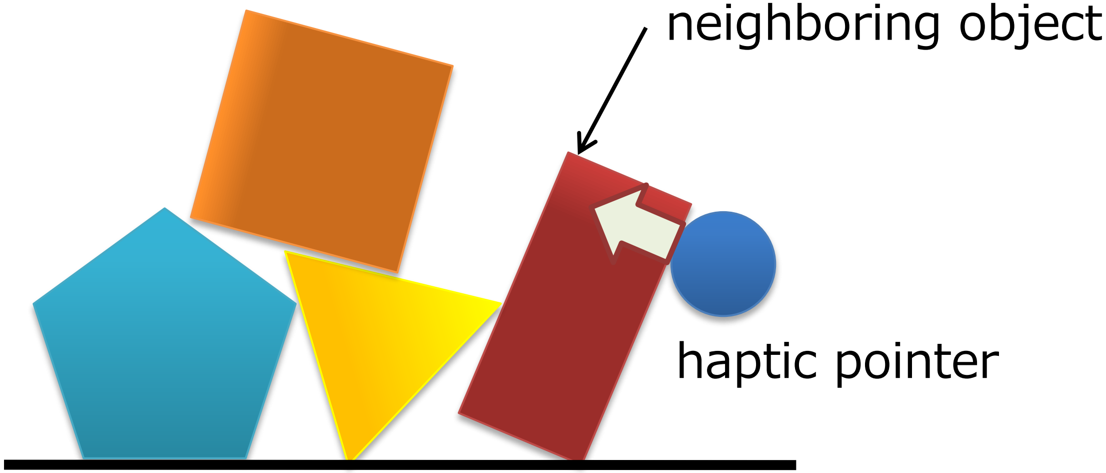
- To include all contact forces for local simulation
  - Pre-simulation with perturbation force and predicting accelerances of neighboring objects

    “Testing Simulation”
  - Running the local simulation based on prediction



Method

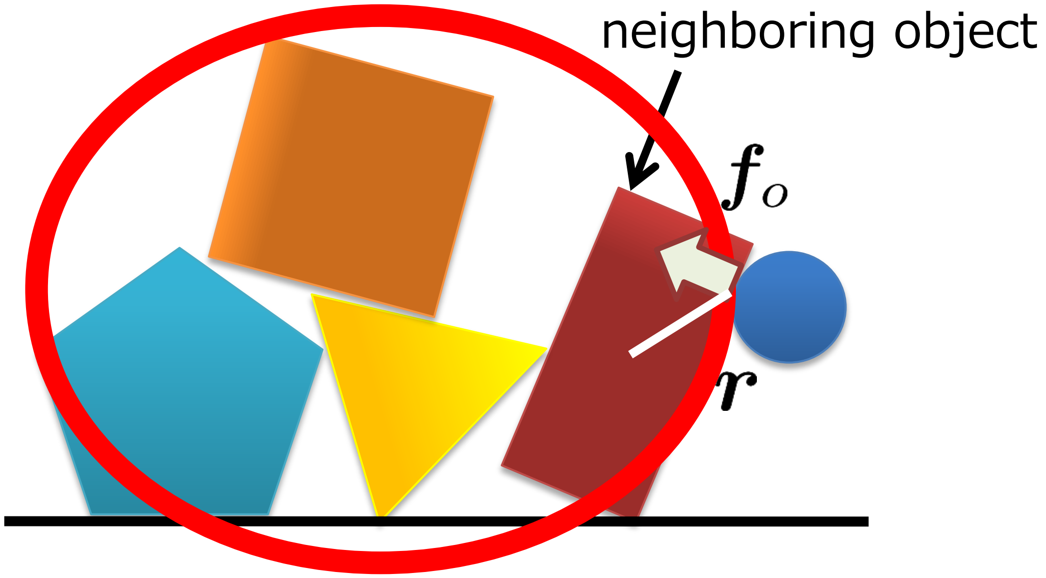
- Selecting the objects nearby the haptic pointer
- Representing the relation between movement of the neighboring object and the force added by the haptic pointer



Motion of the neighboring object

$$\boldsymbol{M}\dot{\boldsymbol{u}} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{K}\int \boldsymbol{u}dt + \boldsymbol{f}_e = \begin{pmatrix} \boldsymbol{f}_o \\ \boldsymbol{r} \times \boldsymbol{f}_o \end{pmatrix}$$

$$\dot{\boldsymbol{u}} = \boldsymbol{A}\boldsymbol{f}_o + \boldsymbol{b}$$



$\boldsymbol{M}, \boldsymbol{B}, \boldsymbol{K}$: mechanical impedance matrix

$\boldsymbol{u}$: velocity and angular velocity of a neighboring object

$\boldsymbol{f}_o$: force added by the haptic pointer

$\boldsymbol{f}_e$: external force like gravity

$\boldsymbol{r}$: point of application of the force $\boldsymbol{f}_o$

Accelerance Matrix

$$\dot{\boldsymbol{u}} = \boldsymbol{A}\boldsymbol{f}_o + \boldsymbol{b}$$

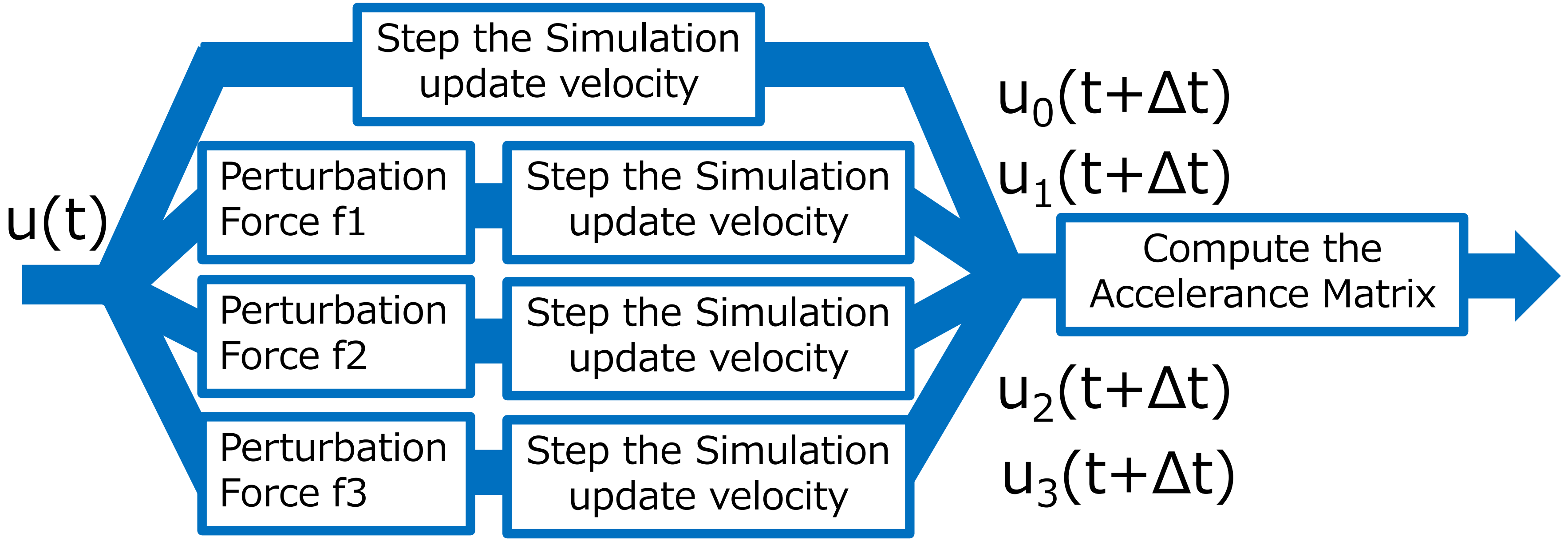
accelerance matrix ($\in\mathbb{R}^{6\times3}$)

acceleration term($\in\mathbb{R}^{6}$)

- A matrix which transforms force into acceleration
  - features inverse mass-inertia matrix
  - the motion can be computed as linear model

Compute A, b by the testing simulation

Testing Simulation



$u(t)$

Step the Simulation update velocity

Perturbation Force f1

Step the Simulation update velocity

Perturbation Force f2

Step the Simulation update velocity

Perturbation Force f3

Step the Simulation update velocity

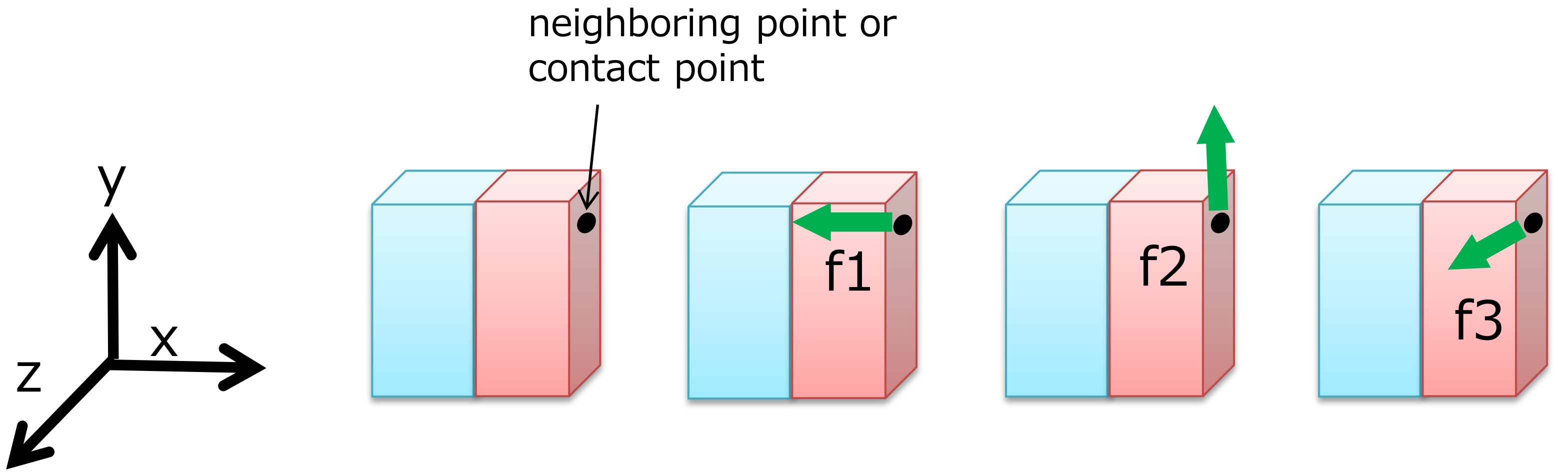
$u_0(t+\Delta t)$

$u_1(t+\Delta t)$

$u_2(t+\Delta t)$

$u_3(t+\Delta t)$

Compute the Accelerance Matrix



neighboring point or contact point

Computation of Accelerance Matrix

- Difference equation of motion

$$\boldsymbol{u}(t+\Delta t) = \boldsymbol{u}(t) + (\boxed{\boldsymbol{A}}\boldsymbol{f} + \boxed{\boldsymbol{b}})\Delta t$$

known

$$\boldsymbol{b} = \frac{\boldsymbol{u}_0 - \boldsymbol{u}_{(t)}}{\Delta t}$$

$$\begin{cases} \boldsymbol{A}\boldsymbol{f}_1 & = (\boldsymbol{u}_1 - \boldsymbol{u}_{(t)})\Delta t^{-1} - \boldsymbol{b} \quad (\equiv \boldsymbol{y}_1) \\ \boldsymbol{A}\boldsymbol{f}_2 & = (\boldsymbol{u}_2 - \boldsymbol{u}_{(t)})\Delta t^{-1} - \boldsymbol{b} \quad (\equiv \boldsymbol{y}_2) \\ \boldsymbol{A}\boldsymbol{f}_3 & = (\boldsymbol{u}_3 - \boldsymbol{u}_{(t)})\Delta t^{-1} - \boldsymbol{b} \quad (\equiv \boldsymbol{y}_3) \end{cases}$$

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \boldsymbol{y}_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_1 & \boldsymbol{f}_2 & \boldsymbol{f}_3 \end{bmatrix}^{-1}$$

Local Dynamics Simulation

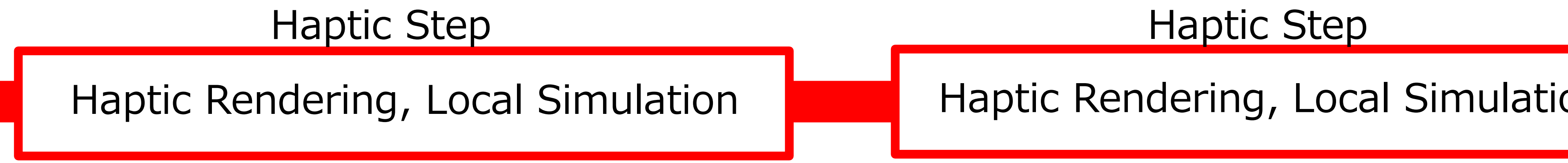
In the haptic step, after the haptic rendering

- Using the accelerance matrix and rendered force

$$\boldsymbol{u}(t+\Delta t)=\boldsymbol{u}(t)+(\boldsymbol{A}\boldsymbol{f}+\boldsymbol{b})\Delta t$$

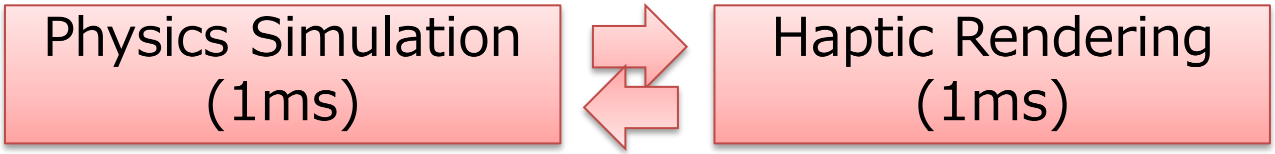
current velocity
angular velocity

rendered force


Haptic Step
Haptic Rendering, Local Simulation
Haptic Step
Haptic Rendering, Local Simulatio

Methods for Comparison

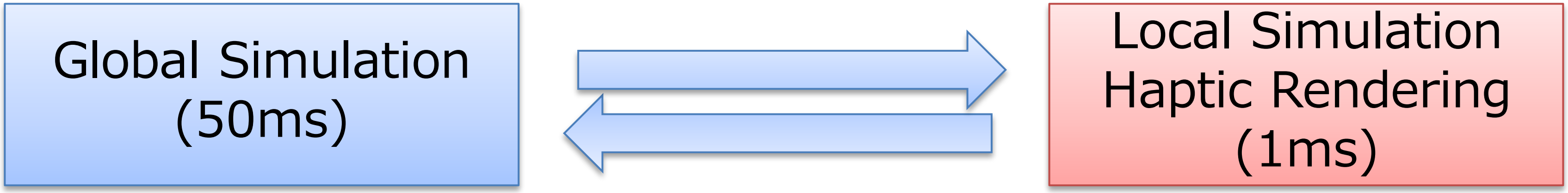
- Method 1 : Updates simulation with high cycle



Physics Simulation (1ms)

Haptic Rendering (1ms)

- Method 2 : Using local dynamics simulation

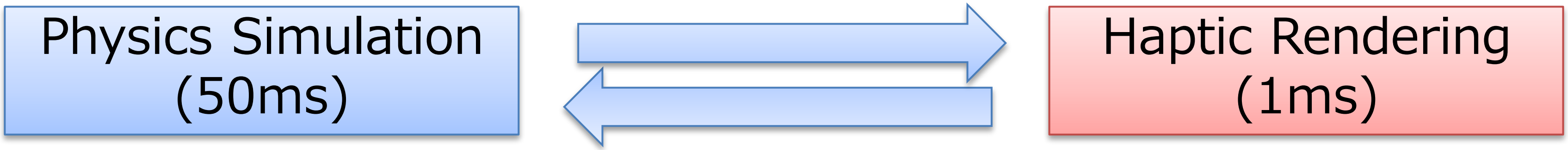


Global Simulation (50ms)

Local Simulation Haptic Rendering (1ms)

- Method 3 :Using impulse communication

[Hasegawa *et al.*]

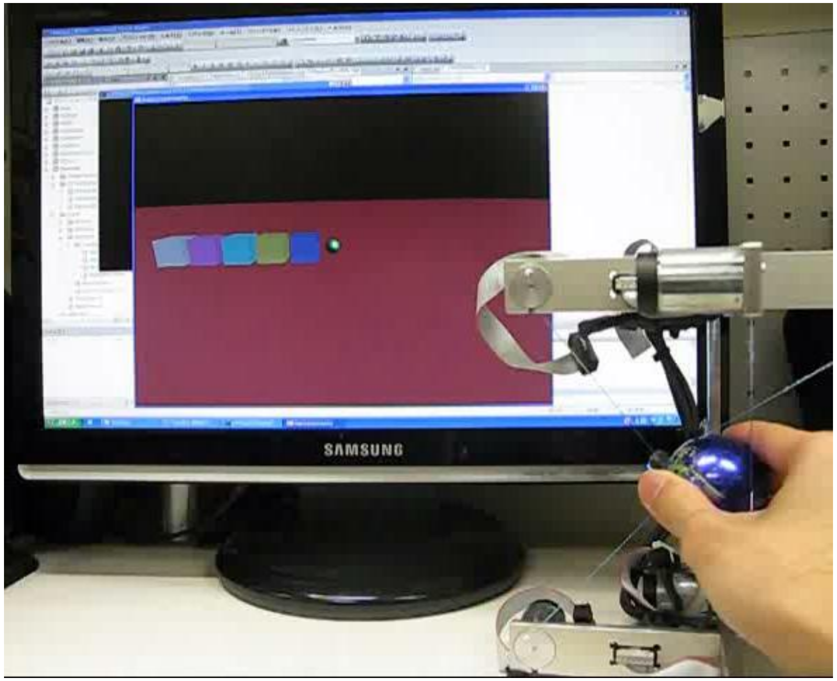


Physics Simulation (50ms)

Haptic Rendering (1ms)

Comparison of the Computation Time

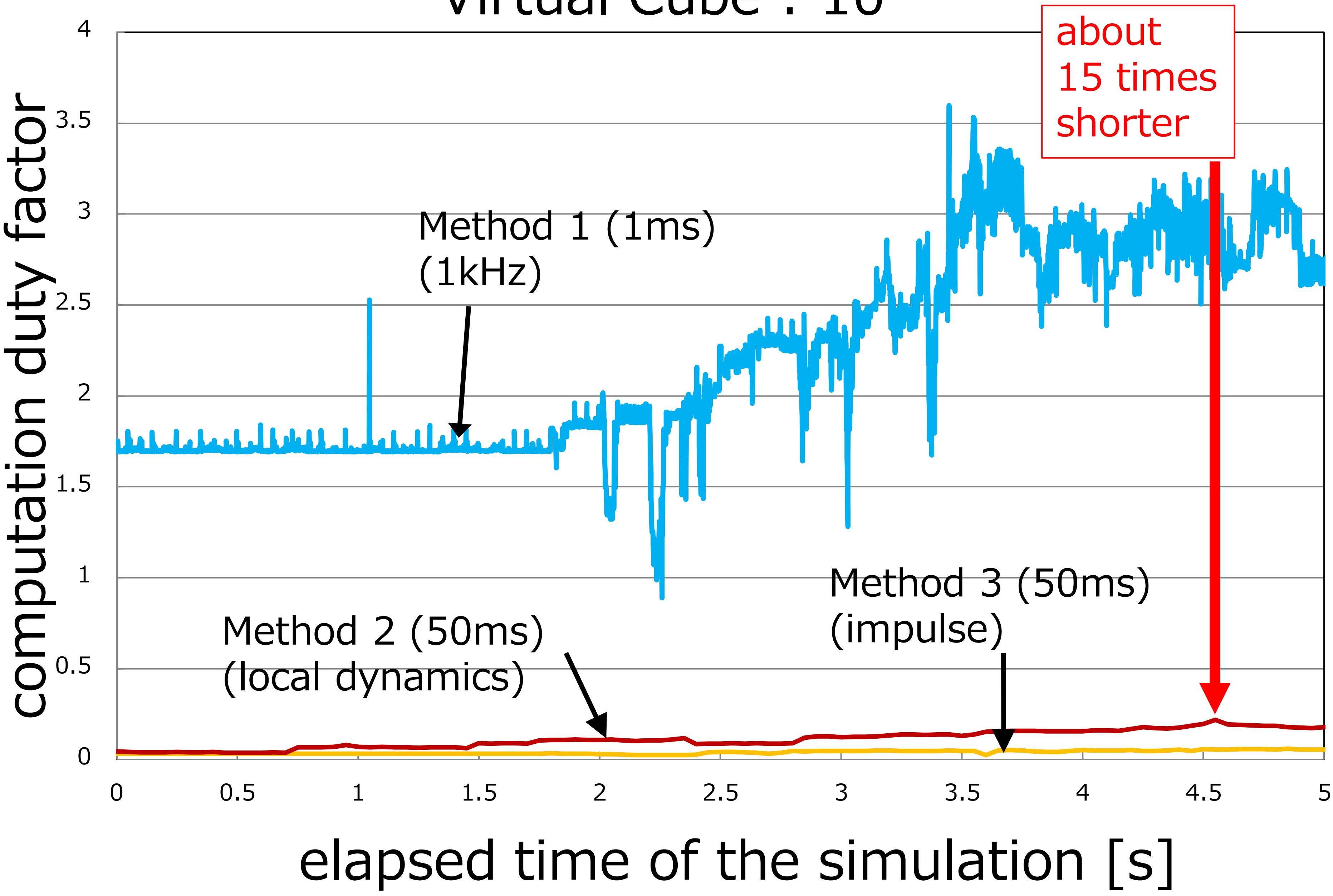
- Measuring computation time of physics simulation
  - one step of the simulation
- Situation
  - Pushing the virtual cubes with the haptic pointer
  - 10 virtual cubes
- Indicators for comparison



$$\text{Computation Duty Factor} = \frac{\text{Computation Time}}{\text{Time Step}}$$

*computation duty factor <= 1 means completed the simulation in real time

Virtual Cube : 10

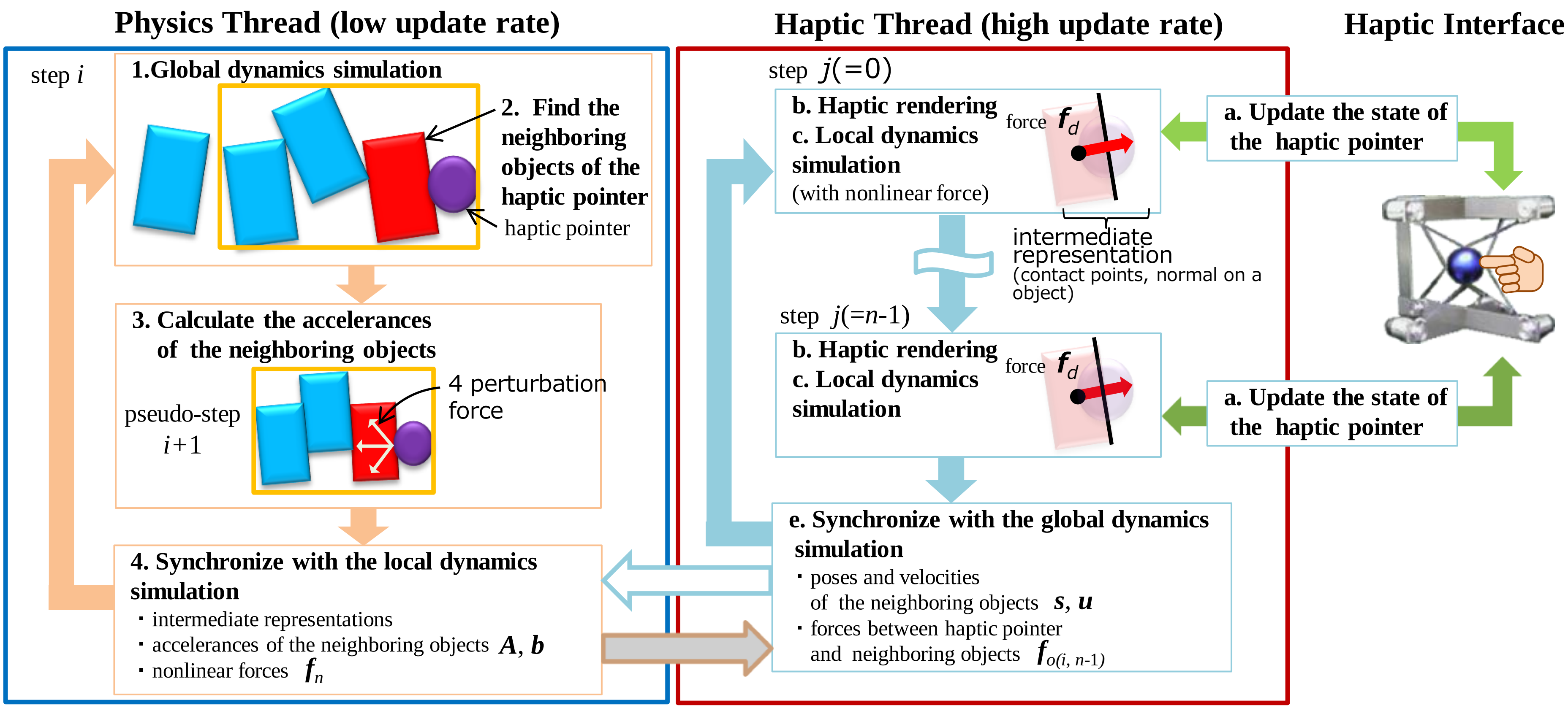


Summary and Future Work

- Parallel dynamic simulation for haptic interaction
- Accelerance matrix can include all contact forces for local simulation

- Extend to 6-DoF force feedback
- Support deformable objects

Thank you !



Physics Thread (low update rate)

step i

1.Global dynamics simulation

2. Find the neighboring objects of the haptic pointer

3. Calculate the accelerances of the neighboring objects

pseudo-step $i+1$

4. Synchronize with the local dynamics simulation

- intermediate representations
- accelerances of the neighboring objects $\boldsymbol{A}, \boldsymbol{b}$
- nonlinear forces $\boldsymbol{f}_n$

Haptic Thread (high update rate)

step $j(=0)$

b. Haptic rendering

c. Local dynamics simulation

(with nonlinear force)

intermediate representation (contact points, normal on a object)

step $j(=n\text{-}1)$

b. Haptic rendering

c. Local dynamics simulation

e. Synchronize with the global dynamics simulation

- poses and velocities of the neighboring objects $\boldsymbol{s}, \boldsymbol{u}$
- forces between haptic pointer and neighboring objects $\boldsymbol{f}_{o(i,\, n\text{-}1)}$

Haptic Interface

a. Update the state of the haptic pointer

a. Update the state of the haptic pointer