

# 解析法の物理シミュレータのための 局所的な動力学計算を行う力覚レンダリング

須佐 育弥<sup>\*1</sup>

長谷川 晶一<sup>\*1</sup>

Haptic Rendering Including Local Dynamics Calculation  
for Rigid Body Simulator Based on Analytical Methods

Ikumi Susa<sup>\*1</sup> Shoichi Hasegawa<sup>\*1</sup>

**Abstract** – In this paper we present a method for displaying force feedback using rigid body simulator based on analytical methods. A force feedback system consists of haptic rendering process and physical simulation process. Since stable control of haptic interfaces requires update rate of higher than 1kHz, haptic rendering processes require the same update rate. On the other hand, rigid body simulators require low update rate to run simulation because of computational limitation. Therefore, it is difficult to display correct force feedback to users. We present a synchronization technique between physical simulation process and haptic rendering process using local dynamics. This technique enable to make consistency between haptic rendering process and physical simulation process. We did a simple simulation to check the effectiveness of the synchronization technique.

**Keywords** : haptic rendering, force feedback, physical simulation

## 1. はじめに

ユーザがVR(バーチャルリアリティ)世界とリアルタイムにインタラクションする方法として、力覚を使用する力覚提示システムがある。力覚提示システムの一つとして力覚インタフェースと物理シミュレータを組み合わせ、ユーザにVR世界の物体を現実の物体を扱うかのように感じさせるシステムがある。このようなシステムではユーザは直感的な操作を実現できることから、設計、教育、訓練、エンタテインメントへの応用が期待されている。

一般に力覚提示システムを使用して、人に力覚を提示する際、人が安定に力覚を感じるためには1kHz以上の更新周波数で力覚インタフェースを制御する必要がある<sup>[1]</sup>。しかし、物理シミュレータの更新周波数が高いと計算回数が増えるため多くの物体をシミュレーションすることができない。

## 2. 関連研究

そこで力覚提示システムの処理をVR世界全体のシミュレーションを行う物理プロセスと力覚レンダリングを行う力覚プロセスに分割し、プロセス間で通信を行う手法が提案されている<sup>[2]</sup>。長谷川ら<sup>[3]</sup>は、物理プロセスの更新周期の間にユーザが入力した力積を力覚プロセスで計算し、その力積をもとに物理プロセスでVR世界を更新する手法を提案している。さらに、力覚プロセスで物体の形状特徴を内挿により補間を行

うことで、ユーザに安定な力覚を提示している。しかし、この手法では力覚を安定に提示することができるが、ユーザの入力をVR世界に反映させる時に遅れが生じる。これによりユーザに提示される形状特徴の更新が遅れるため、ユーザはその物体を重く感じ正確な力覚を感じることができない。

岩下ら<sup>[4]</sup>、大内ら<sup>[5]</sup>は、力覚プロセスで力覚ポイントを中心とした一定の領域内について局所的な動力学計算をする手法を提案している。この手法では力覚ポイントが触れる最近傍物体が高速で更新されているため、ユーザの入力を遅れなく物体の運動に反映することができる。このシステムでは最近傍物体のみを高速に更新しているため、力覚ポイントが触れている最近傍物体に他の物体が接触している場合は、あらかじめ物理プロセスで最近傍物体に対する周囲の制約をもとめ、局所的な動力学計算に使用することで、ユーザに精度の高い力覚を提示することを実現している。

しかし、これらの手法では局所的な動力学計算の範囲外の物体が領域内の物体に衝突した場合に衝突の影響が領域内の物体の運動に反映されないことや、領域内にある物体を通して領域外の物体を押しした場合に、領域外の物体に接触力が加わらず領域外の物体の運動に反映されない。このように、物理シミュレーションと局所的な動力学計算との間で整合性がとれていないためである。

そこで、本稿ではプロセス分割した力覚提示システムにおいて、物理シミュレーションと局所的な動力学計算との間で最近傍物体について速度、位置について

\*1: 電気通信大学

\*1: University of Electro-Communications

整合性をとり，正確な力覚提示が実現できるための手法を提案する．

### 3. 提案手法

#### 3.1 物理プロセスと力覚プロセスの同期処理

ユーザに正確な力覚を感じさせるためには，力覚ポインタの入力をリアルタイムに最近傍物体の運動に反映させなければならない．しかし，現実と同様の振る舞いをさせるためには，物理プロセスで低速に更新されている物体が力覚プロセスで高速に更新されている最近傍物体に対して影響を及ぼす場合，最近傍物体の運動はその影響により変化すべきである．

岩下ら，大内らはVR世界の一部を切り取り局所的な動力学計算を行っていたが，提案手法ではVR世界の一部を複製することで，力覚ポインタについて最近傍の物体のシミュレーションを物理シミュレーションと局所的な動力学計算の両方で行う．そして，これらのシミュレーション間で整合性をとることで，上記の条件を考慮することができると考えられる．そこで，物理シミュレーションと局所的な動力学計算の間で整合性をとる同期処理手法を以下に提案する．

(1) 力覚レンダリングでは力覚ポインタと物体の間の微小な侵入量に基づいて力を計算しているため，刻み時間が大きい物理プロセスの最近傍物体の位置を刻み時間が小さい力覚プロセスに反映させると，ユーザに予期しない力覚を提示してしまう．そこで，高速で更新する力覚プロセスの最近傍物体の位置を物理プロセスの最近傍物体に直接反映させる．

(2) ユーザの入力により運動が変化する最近傍物体及び最近傍物体に接触する物体がある場合，最近傍物体の運動は接触している物体からの接触力を考慮しながらリアルタイムに計算されなければならない．そこであらかじめ予測シミュレーションにより最近傍物体の運動係数を決定し，局所的な動力学計算で使用することで，最近傍物体について接触する物体の接触力を考慮した動力学計算を行う．

(3) 予測シミュレーション対象外の物体が衝突により最近傍物体に影響を与えた場合は，衝突による力を物理プロセスで計算し，力覚プロセスに送信し力積として最近傍物体の運動に反映する．ユーザへの衝突による力の提示には時間遅れが発生するが，物体同士の衝突のタイミングを知覚するのは視覚であり，視覚は60Hz程度の時間分解能しかないため，衝突によるユーザへの力覚提示は遅れても支障はないといえる<sup>[6]</sup>．

これらを行うことで物理プロセス，力覚プロセスの最近傍物体は整合性を実現し，ユーザに正確な力覚提示ができる．

#### 3.2 システム構成

各プロセス間で通信すべき情報をまとめると以下のようなになる．

物理プロセスから力覚プロセスへ

力覚ポインタ最近傍の物体の情報 力覚プロセスへの局所的な動力学計算に必要な物理情報は最近傍物体の質量，慣性モーメント，重心位置，位置，姿勢，速度，角速度，力覚ポインタと最近傍物体の接触点である．

最近傍物体の運動係数 3.1節(2)及び4.1節参照

最近傍物体が衝突によって受けた力積 3.1節(3)参照

力覚プロセスから物理プロセスへ

最近傍物体の位置，姿勢 3.1節(1)参照

力覚ポインタが加えた力積 最近傍物体に他の物体が接触している状態でユーザの入力により，接触物体の運動が変化する場合，接触している物体の運動を処理するのは物理プロセスである．そのため，ユーザが入力した力積を物理プロセスに送信し，接触する物体を力積をもとに更新する．

力覚ポインタの位置，姿勢 VR世界上の最近傍物体を探索する基準や最近傍物体との接触点の計算で使用する．

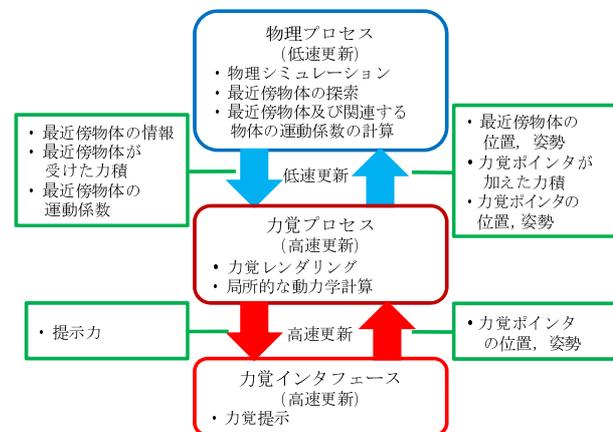


図1 システム構成図

Fig.1 System configuration

上述のシミュレーション間で整合性を取るための提案手法を実現するためのシステムの構成図を図1に示す．また以下に各プロセスの処理手順について説明する．

#### ● 物理プロセス

1. 力覚プロセスから情報を受信
2. 局所的な動力学計算の結果を反映 (6.1節)  
受信した情報を最近傍物体と最近傍物体に接触する物体に反映する．
3. 物理シミュレーション

最近傍物体を除く VR 世界全体の物理シミュレーションを行う。

4. 最近傍物体の探索  
力覚ポイントに対して最近傍の物体を探索する。
5. 最近傍物体と最近傍物体に接触する物体の運動係数の決定 (4.1 節)  
予測シミュレーションにより、力覚ポイントの入力によって運動が変化する物体の運動係数を求める。
6. 力覚プロセスへ情報を送信  
最近傍物体が受けた力積と最近傍物体の運動方程式を力覚プロセスに送信する。

● 力覚プロセス

1. 物理プロセスから情報を受信
2. 物理シミュレーションの結果の反映 (6.2 節)  
受信した情報を最近傍物体に反映する。
3. 力覚ポイントの位置の更新  
力覚インタフェースから力覚ポイントの位置を受信する。
4. 力覚レンダリング (節 5.1)  
提示力の計算をし、力覚インタフェースに提示力を送信する。
5. 局所的な動力学計算 (節 5.2)  
力覚ポイントが最近傍物体に加えた提示力と、物理プロセスで求めた運動係数に基づいて、最近傍物体を更新する。
6. 3, 4, 5 の処理を  $n$  回行う  
物理プロセスが終わるまで 3, 4, 5 の処理を繰り返す。
7. 物理プロセスに情報を送信

3.3 プロセス間の通信

物理プロセスと力覚プロセスの同期は物理プロセスのステップが終わる時点で行う。物理プロセスの更新周期に対する力覚プロセスの更新周期の比率を  $n$  とすると、力覚プロセスステップを  $n$  回行っている間に物理プロセスステップを 1 回行い、プロセス間通信をする。このとき、 $i$  を第  $i$  番目の物理プロセスステップ、 $j$  を第  $i$  番目の物理プロセスステップを行っている間の第  $j$  番目の力覚プロセスステップとする。ただし、 $0 \leq j < n$  で  $j$  は物理プロセスと同期後に 0 となる。

4. 物理プロセスの処理

4.1 局所的な動力学計算のための運動係数の決定

図 2 のように最近傍物体にその他の物体が接触している状況を考える。3.1 節 (2) で述べたように、ユーザが操作する力覚ポイントで最近傍物体を押したと

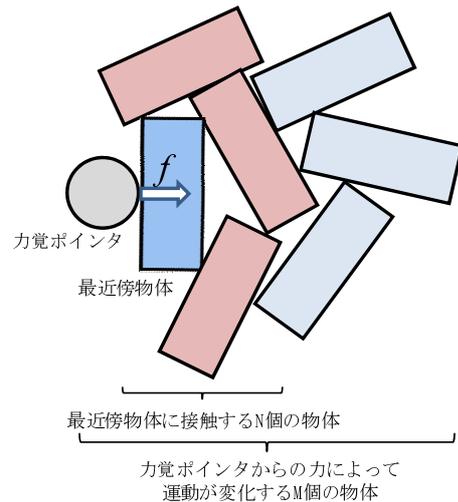


図 2 最近傍物体に接触する物体

Fig.2 Objects contact with nearest object

き、力覚ポイントには接触する物体を含めた慣性が伝わり、ユーザに提示されなければならない。そこで予測シミュレーションをあらかじめ物理プロセスで行い、最近傍物体の運動係数を計算し、局所的な動力学計算で使用する。

第  $i-1$  番目の物理プロセスステップで力覚ポイントと最近傍物体の接触点に仮に力  $f$  を加えたとき、運動が変化する  $M$  個の物体の運動方程式は

$$\dot{u}_{(i-1)} = A_{(i-1)}f + b_{(i-1)} \quad (1)$$

である。ただし、 $\dot{u}_{(i-1)}$  は  $M$  個の物体の加速度、角加速度、 $A_{(i-1)}$  は  $6M \times 3$  の係数行列、 $b_{(i-1)}$  は重力などの外力である。ここで  $f$  を  $(1, 0, 0)^t$ ,  $(0, 1, 0)^t$ ,  $(0, 0, 1)^t$ ,  $(0, 0, 0)^t$  とし、それぞれ物理シミュレーションを 1 ステップ進めると、関連する物体  $M$  個の 1 ステップ後の速度、角速度が得られる。式 (1) を差分方程式に変形すると、

$$u_{(i)} = u_{(i-1)} + (A_{(i-1)}f + b_{(i-1)})\Delta t_p \quad (2)$$

となることから、得られた速度、角速度を代入すると  $A_{(i-1)}$  と  $b_{(i-1)}$  が得られる。得られた  $A_{(i-1)}$ ,  $b_{(i-1)}$  の最近傍物体の部分  $A_{p(i-1)}$ ,  $b_{p(i-1)}$  とし第  $i$  番目の力覚プロセスステップに送信する。

5. 力覚プロセスの処理

5.1 力覚レンダリング

力覚ポイントの位置と最近傍物体の情報からユーザへの提示力の計算を行う。提示力は力覚ポイントが物体に侵入した場合に、力覚ポイントと物体を仮想のばねダンパモデルから計算する。ここでは力覚ポイントが物体に侵入した量とその微分に比例した値を提示力

としている．ユーザへの提示力  $f_{d(i,j)}$  は

$$f_{d(i,j)} = kx_{h(i,j)} + d\dot{x}_{h(i,j)} \quad (3)$$

である．ただし， $k$  はばね係数， $d$  はダンパ係数， $x_{h(i,j)}$  は最近傍物体への力覚ポイントの侵入量， $\dot{x}_{h(i,j)}$  は侵入量の微分である．また，物体に加わる力  $f_{h(i,j)}$  は作用反作用の法則より，

$$f_{h(i,j)} = -f_{d(i,j)} \quad (4)$$

となる．

## 5.2 周囲の制約を考慮した局所的な動力学計算

予測シミュレーションによって求めた運動係数  $A$ ， $b$  を局所的な動力学計算で使用する．ユーザが実際に力覚ポイントで加えた力を運動方程式に代入し，力覚プロセスの最近傍物体の速度，位置の更新を行う．第  $(i, j)$  番目の最近傍物体の運動方程式は

$$\dot{u}_{h(i,j)} = A_{p(i-1)} f_{h(i,j)} + b_{p(i-1)} \quad (5)$$

となる．これより，

$$u_{h(i,j)} = u_{h(i,j-1)} + (A_{p(i-1)} f_{h(i,j)} + b_{p(i-1)}) \Delta t_h \quad (6)$$

に従って速度，角速度を更新を行う．続けて位置，姿勢の更新を行う．ただし， $u_{h(i,j)}$  は力覚プロセスでの最近傍物体の速度， $f_{h(i,j)}$  はユーザが実際に入力した力， $\Delta t_h$  は局所的な動力学計算の刻み時間である．

力覚プロセスステップを  $n$  回行った後，最近傍物体の位置，姿勢，ユーザの入力した力積の総和  $F_{h(i)}$  を  $i+1$  番目の物理プロセスステップへ送信する．力積の総和  $F_{h(i)}$  は

$$F_{h(i)} = \sum_{j=0}^{n-1} f_{h(i,j)} \Delta t_h \quad (7)$$

である．

## 6. プロセスの同期処理

### 6.1 物理プロセスへのユーザの入力の反映

第  $i+1$  番目の物理プロセスステップにおいて，物理シミュレーション更新時に第  $i$  番目の力覚プロセスステップの結果を反映させる．結果の反映対象は力覚ポイント最近傍の物体とそれに接触する  $M$  個の物体である．第  $i-1$  番目の物理プロセスステップの予測シミュレーションで求めた運動係数行列  $A_{(i-1)}$  と力積の総和  $F_{h(i)}$  を物理シミュレーションの刻み時間  $\Delta t_p$  で割ると，

$$F_{p(i)} = A_{(i-1)} \frac{F_{h(i)}}{\Delta t_p} \quad (8)$$

関連する  $M$  個の物体に加わる力  $F_{p(i)}$  が得られる．力  $F_{p(i)}$  を運動方程式に組み込み，速度，角速度の更新を行う．続けて，位置，姿勢の更新を行う．

ここで，物理プロセスの最近傍物体の位置，姿勢は第  $(i, n-1)$  番目の力覚プロセスステップの最近傍物体の位置，姿勢に置き換える．位置，姿勢を置き換えることにより，ユーザの入力を正確に物理シミュレーションに反映できる．

### 6.2 力覚プロセスへの衝突による力の反映

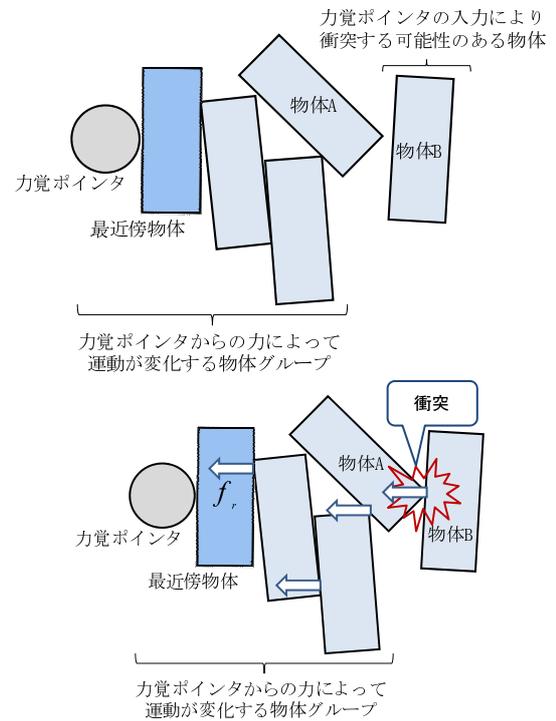


図3 最近傍物体への衝突の影響  
Fig.3 Collision effects on nearest object

例として図3のような状況で力覚ポイントで物体を押し続けていくと，物体A，Bが衝突する．局所的な動力学計算では最近傍物体に接触している物体に対しては周囲の制約として考慮されているが，新たに衝突が起きることが想定されていないため，衝突が起きた場合の力が局所的な動力学計算へ反映されない．提案手法では物理シミュレーションの衝突応答で，最近傍物体が衝突により受ける力を力覚プロセスを  $n$  回行ううちの1回目に反映させ，速度の整合性をとる．例えば，第  $i-1$  番目の物理プロセスステップでの物理シミュレーションの衝突応答で最近傍物体が衝突により力  $f_{r(i-1)}$  を受けた場合，以下の式により第  $(i, 0)$  番目

の力覚プロセスステップへ反映する．

$$\begin{aligned} \mathbf{u}_{h(i,0)} = & \mathbf{u}_{h(i-1,n-1)} \\ & + (\mathbf{A}_{p(i-1)} \mathbf{f}_{h(i,0)} + \mathbf{b}_{p(i-1)}) \Delta t_h \quad (9) \\ & - \left( \begin{array}{c} \frac{\mathbf{f}_{r(i-1)}}{m_c} \\ \mathbf{I}_c^{-1} (\mathbf{r}_{(i-1)} \times \mathbf{f}_{r(i-1)}) \end{array} \right) \Delta t_p \end{aligned}$$

ただし， $m_c$ ， $\mathbf{I}_c^{-1}$  は最近傍物体の質量と慣性逆行列， $\mathbf{r}_{(i-1)}$  は最近傍物体の重心からみた衝突点の位置である．

## 7. 評価・考察

提案手法の有効性を評価するために以下の構成で実装した．

力覚インタフェース

SPIDAR-G6<sup>[7]</sup>

計算機

Windows Vista Business

Intel(R) Core(TM)2 CPU U7600 1.20GHz

プロセス更新周期

物理プロセス 20Hz

力覚プロセス 1kHz

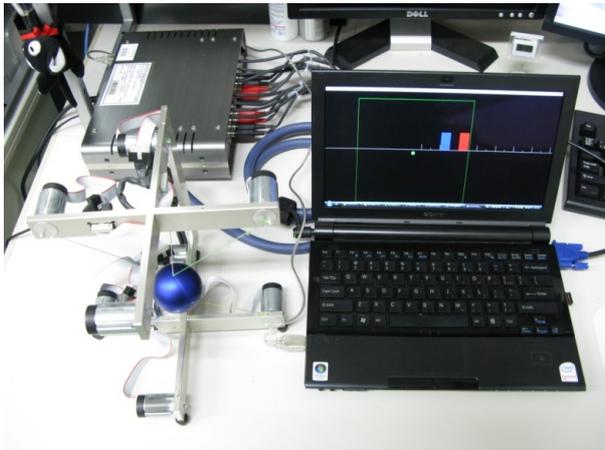


図4 実験構成

Fig. 4 Consist of experiment

### シミュレーション 1

提案手法により物理プロセスの最近傍物体と力覚プロセスの最近傍物体の速度について整合性がとれているかを確認するために，物体を2個並べ，力覚ポイントで最近傍物体を0.5Nで押し続け隣合う物体と衝突させるシミュレーションを行った．シミュレーションでは両物体の質量を1kg，摩擦係数0，反発係数0，物体間の距離を50cmとし，力覚ポイントが触れる最近傍物体の速度を計測した．

#### シミュレーション 1 結果・考察

シミュレーション結果を図5に示す．衝突時刻までは力覚ポイントからの力により，力覚プロセスの最近

傍物体の速度が変化し，速度の変化分に応じて物理プロセスの最近傍物体の速度に反映している．衝突時刻で隣合う物体と衝突し，衝撃力により，最近傍物体の速度が低下している．衝突が発生した時刻付近を拡大した図では，まず物理プロセスに存在する最近傍物体と隣合う物体の衝突により，物理プロセスの最近傍物体の速度が変化している．次に力覚プロセスの最近傍物体が物理プロセスの最近傍物体の速度の変化分を力積として受信し，物理プロセスで起きた衝突による運動の変化を反映させている．衝突直前で物理プロセスの最近傍物体と力覚プロセスの最近傍物体の速度は等しくない．これは物理プロセスで衝突による運動の変化と力覚プロセスの力覚ポイントからの力積の総和によって衝突後の速度が計算されたためである．しかし，衝突による運動の変化が力覚プロセスで反映されることによって物理プロセスの最近傍物体の速度と等しくなっていることから，整合性はとれているといえる．これより，力覚プロセスから物理プロセスへ，物理プロセスから力覚プロセスへの最近傍物体の運動の変化が反映され，速度の観点から整合性がとれているといえる．

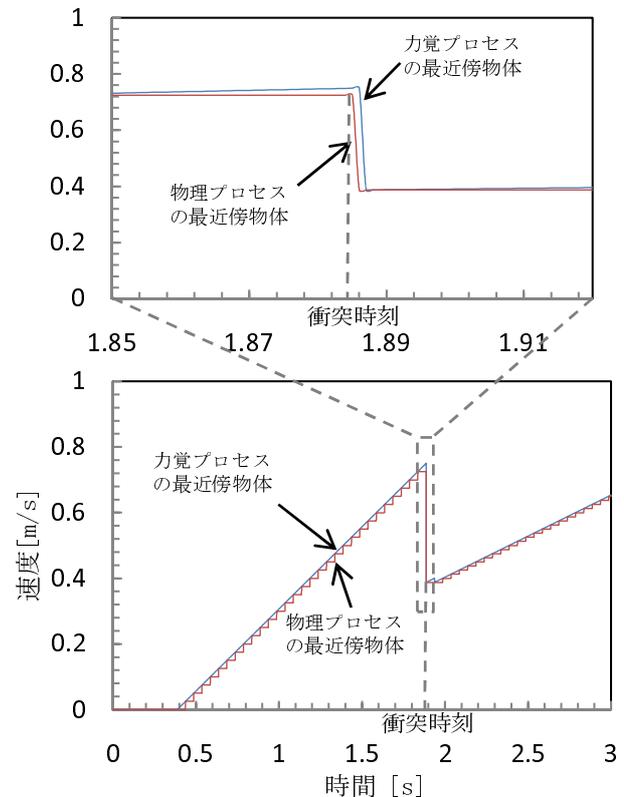


図5 シミュレーション 1 結果

Fig. 5 Consistency of simulation

### シミュレーション 2

従来の手法に対して提案手法の有効性を評価するためのシミュレーションを行った．シミュレーション 1

と同様の構成で力覚ポイントが押す最近傍物体の位置を測定した。このシミュレーションを提案手法、周囲の制約を考慮する局所的な動力学計算のみを行う大内らの手法<sup>[5]</sup>、最近傍物体を物理シミュレーションで更新する長谷川らの手法<sup>[3]</sup>、プロセス分割をせずに全てを1kHzの更新周期で更新する方法で行った。提案手法、大内らの手法、長谷川らの手法とプロセス分割をしている手法は更新周波数を1kHz-20Hzに設定した。

シミュレーション2 結果・考察

図6は各手法で行った位置を表すグラフである。力覚ポイントの入力により、最近傍物体の位置が変化する。位置変化は隣合う物体と衝突をするまで、長谷川らの手法を除く手法について同様の軌跡をとっている。長谷川らの手法がその他の手法の軌跡と異なるのは力覚ポイントの入力が物体の位置変化に反映されるまでに時間遅れがあるためである。隣合う物体と衝突する時、大内らの手法は最近傍物体に対して接触力が加わらないため、それによる加速度変化がないのでその他の方法と軌跡が異なっている。提案手法は長谷川らの手法、全てを1kHzで更新する方法と比べ、衝突による位置変化が遅れていることがわかる。この時間遅れは提案手法で他物体による衝撃力を低速更新の物理プロセスで計算してから、高速更新の力覚プロセスに反映させる時の遅れである。本研究ではユーザの入力を正確にVR世界に反映し、かつ安定な力覚提示を行うことを目的としているので、この時間遅れによる問題はないといえる。これより、提案手法は従来のプロセス分割の手法と比べ、全てを高更新周期で行う手法に近いといえる。

## 8. 結論

プロセス分割をした力覚提示システムにおいて、物理シミュレーションと局所的な動力学計算の間で速度、位置の整合性をとることで、ユーザへの正確な力覚提示、ユーザが入力した力を正確に物理シミュレーションに反映する手法を提案した。さらに、提案手法を従来手法とシミュレーションで比較し、有効性を確認した。

## 参考文献

- [1] Lonnie Love, Wayne Book: Contact Stability Analysis of Virtual Walls; Proc. of Dynamic Systems and Control Division, pp.689-694, (1995).
- [2] Y. Adachi and T. Kumano and K. Ogino: Intermediate representation for stiff virtual objects; Proc. IEEE Virtual Reality Annual International Symposium, pp.203-210 (1995).
- [3] 長谷川晶一: 動的な仮想世界の力覚提示のためのプロセス間通信; 電子情報通信学会論文誌 D-2, Vol.J82-D-2, No.10, pp.1-8 (1999).

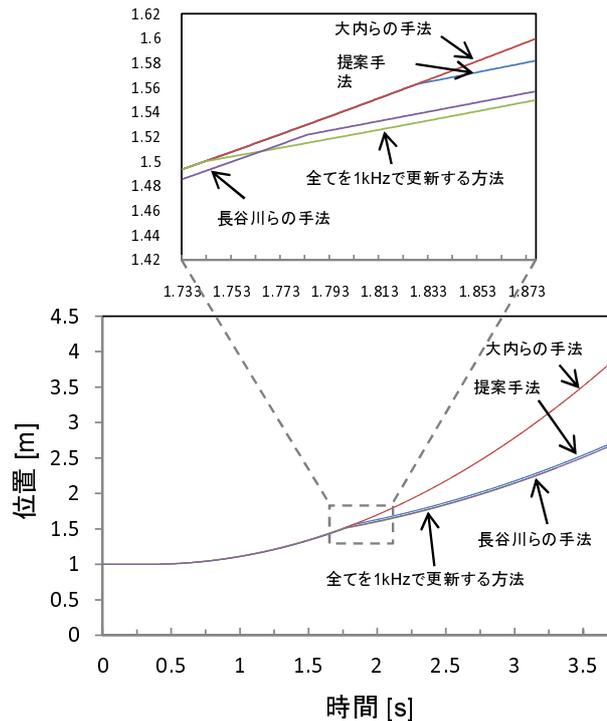


図6 シミュレーション2 結果  
Fig.6 Compare with other methods

- [4] 岩下克, 赤羽克仁, 長谷川晶一, 小池康晴, 佐藤誠: 局所的な動力学計算を行う力覚レンダリング; 電子情報通信学会技術研究報告. MVE, マルチメディア・仮想環境基礎, Vol.105, No.106, pp.73-78 (2005).
- [5] 大内政義, 長谷川晶一, 小池康晴, 佐藤誠: 物理シミュレータ上での力覚提示のための局所的な動力学計算を行う力覚レンダリング; 電子情報通信学会技術研究報告. MVE, マルチメディア・仮想環境基礎, Vol.106, No.611, pp.1-6 (2007).
- [6] 日本視覚学会: 視覚情報処理ハンドブック; 朝倉書店, pp.219-220 (2000).
- [7] 佐藤誠, 平田幸広, 河原田弘: 空間インタフェース装置 SPIDAR の提案; 電子情報通信学会論文誌, Vol.J74-D-2, No.7, pp.887-894 (1991).