

## 6自由度力覚レンダリングのための中間表現

須佐 育弥<sup>\*1</sup> 長谷川 晶一<sup>\*2</sup>

### Intermediate Representation for 6-DoF Haptic Rendering

Ikumi Susa<sup>\*1</sup> and Shoichi Hasegawa<sup>\*2</sup>

**Abstract** – In this paper, we propose a novel intermediate representation to realize multi-rate 6-DoF haptic display system. Our proposed intermediate representation has various features. Use of the intermediate representation can display haptic feedback without inertial effect of haptic pointer. And polygon models are available for both haptic pointer and virtual objects without any pretreatment. In order to respond to 6-DoF operation, the intermediate representation consists of a plane and vertices. For haptic rendering, we have employed Yokoyama’s constraint-based contact response method to display stable torque. Moreover, we attached a friction cone to each vertex of intermediate representation to calculate Coulomb’s frictional force. Finally, we investigated effectiveness of the proposed method.

**Keywords** : 6-dof haptic rendering, force feedback, intermediate representation, multi-rate, friction

#### 1 はじめに

人間がバーチャル世界に対してインタラクションする方法の一つとして、力覚インタラクションがある。力覚インタラクションはバーチャル世界に存在する物の形状、重さ、硬さ等の物理的な情報を力として人間に伝えることを可能にした。力覚インタラクションは設計、訓練、娯楽等、様々な分野での応用が期待され、研究が盛んに行われている。しかしながら、力覚インタラクションの方法自体に多くの研究課題があり、上記のような分野での実用化は容易ではない。

例えば力覚インタラクションの研究の一つとして、物理シミュレーションされたバーチャル世界との力覚インタラクションがある。物理法則に従ってバーチャル世界が運動することにより、バーチャル世界でも実世界で人間が行う物理的なインタラクションと同様のことができる。また、人間は普段の生活において6自由度の操作をし、6自由度の力覚を得ている。そのため、力覚インタラクションにおいても直感的な操作の実現には、6自由度の力覚提示が必要である。

本論文では剛体のバーチャル世界に対する6自由度の力覚レンダリングの手法について提案する。本手法の特徴は、

- 力覚ポインタ（ユーザがバーチャル世界内で操作をする物体）が無拘束の時には力覚が提示されない

- 前処理なしの任意のポリゴンが利用できる（ただし、凹形状の場合は凸分割が必要）
- クーロンの摩擦法則に基づいた摩擦力が提示可能
- 計算量を分散させるためのマルチレートシステムで利用可能

である。提案手法は、デジタルモックアップのための繊細な作業や、計算量を必要とするバーチャル世界（関節で繋がれた鎖やバーチャルクレーン[18]等）とのインタラクションへの応用が期待できる。本論文では第2章で力覚提示システム及び力覚レンダリングの手法について関連研究を紹介する。第3章で研究の目的、提案手法の概要を、第4章で提案手法の実装について説明する。第5章では提案手法を実装したシステムの評価について記し、考察を述べる。最後に第6章で結論を記す。

#### 2 関連研究

物理シミュレーションされたバーチャル世界に対して力覚インタラクションを実現する場合、力覚提示の品質を重視するか、バーチャル世界の複雑さを重視するかでシステム全体の構成が変わり、力覚レンダリングの方法も変わる。本章では力覚提示システムと力覚レンダリングに関する研究を紹介する。

##### 2.1 力覚提示システム

従来の研究から力覚提示システムを構成するための指標が幾つか挙げられている。

- 力覚インタフェース、レンダリングの高速更新 Colgate ら [6] は力覚提示の安定性と更新周期の関係を式として表し、更新周期を高くするほど、安定に提示可能な力の領域が広くなり、高硬度な力覚を提示できることを示している。また、安

<sup>\*1</sup>東京工業大学 総合理工学研究科 知能システム科学専攻

<sup>\*2</sup>東京工業大学 精密工学研究所

<sup>\*1</sup>Department of Computational Intelligence and Systems Science, Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology

<sup>\*2</sup>Precision and Intelligence Laboratory, Tokyo Institute of Technology

定な硬さの提示の指標として 1kHz 以上の更新 [13] が挙げられている。

#### ● 物理シミュレーションの低速更新 [1][16]

大量の物体や柔軟体等の運動は膨大な計算量を必要とする。計算時間を確保するためには、シミュレーションを低速に更新させる必要がある。

これらの要件は相反するもので、バランスを取りながら両立させる必要がある。例えば力覚提示の品質を重視すると、物理シミュレーションの更新を速くする必要があり、シミュレーション 1 ステップに割けられる計算時間は短くなる。そのため、膨大な計算量を必要とする複雑なバーチャル世界を実現することは難しい。一方、複雑なバーチャル世界を重視すると、力覚レンダリングの更新を遅くする必要があり、提示力の品質は落ちる。

以上のような要件を実現するために、それぞれの処理を異なるスレッドに分割し、実行、同期をさせるマルチレートシステムが提案されている。Adachi ら [1], Mark ら [14], Hasegawa ら [8] は力覚レンダリングに必要なバーチャルな物体の形状特徴を中間表現（面法線と点などで表される幾何情報）として表し、スレッド間の通信に利用することでマルチレートシステムを実現している。Otaduy ら [21], Akahane ら [2] は力覚インタフェースとバーチャル物体を仮想のバネダンパで結合させるバーチャルカップリング [6] から提示力を計算し、スレッド間の通信に利用する方法を提案している。

## 2.2 力覚レンダリング

インピーダンス型力覚インタフェースを利用する場合、ユーザからの入力は位置姿勢となる。そのため、ペナルティベースのシミュレータ [9][23] を組み合わせたシングルレートシステムに対してはシミュレータが計算する反力を直接力覚として出力することが可能である。しかし、拘束ベースのシミュレータ [3][10] やマルチレートシステムを扱う場合には、提示力の安定性の問題から反力を直接力覚として出力することは望ましくない。マルチレートシステムにおける力覚レンダリングの方法は大きく 2 通りに分類できる。

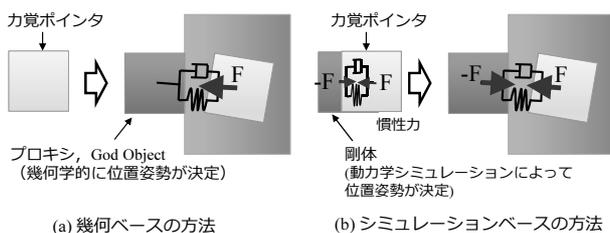


図 1 力覚レンダリングの分類  
Fig. 1 Classification of haptic rendering

## 幾何ベースのレンダリング

力覚ポインタ (力覚インタフェースの操作部分) が物体に侵入している状態から、幾何学的に侵入していない状態 (プロキシ [22], God Object [24]) を求め、プロキシと力覚ポインタ間の距離とバネダンパモデルから提示力をレンダリングする (図 1.(a))。そのため、力覚ポインタが無拘束の状態ではプロキシと力覚ポインタ間の距離は常に 0 となり、ユーザに力覚は提示されない。菊植ら [27] はプロキシを利用した力覚レンダリングの方法に力学的な意味を考え、ユーザの操作を阻害させないダンパの組み込み方法を提案している。幾何ベースのレンダリングは 3 自由度での実装は容易に行えるが、6 自由度での実装はプロキシの位置姿勢を決定するための方法が難しい。

Berkelman ら [4] は多点接触が扱えるように中間表現を拡張し、6 自由度力覚レンダリングを提案している。しかし、この手法では接触点毎の侵入量にバネ係数を乗じたものを力とし、その力の和をユーザへ提示している。そのため、接触点の増加とともに反力が大きくなり、不安定になる。Weller ら [23] は形状を表すメッシュ内部に階層的に球体を配置し、球対球の衝突判定によって求められる侵入量に基づいて力覚レンダリングを行なっている。しかし、エッジが効いたものや硬い面を力覚で表現するのは難しいと考えられる。

## シミュレーションベースのレンダリング

シミュレーションベースのレンダリングはバーチャル物体を力覚ポインタとバーチャルカップリングをさせることで力覚インタラクションを実現する手法である [15][21][2] (図 1.(b))。この方法は物体自体が質量を持ち、物理シミュレーションされるのでユーザは物体の慣性力を受けることになる。しかし、実装が容易であることから本手法を採用している研究は多い。Otaduy ら [21] は物理シミュレーションに陰積分を用いることで、物体の質量を小さくし、慣性の影響を低減させているが完全になくすことはできていない。

Ortega ら [20] は拘束ベースの力覚レンダリング手法を提案している。この手法は力覚ポインタが物体に侵入している時の God Object の拘束加速度と接触していないと仮定した時の非拘束加速度の差から提示力を計算しているが、動力学世界での利用については言及していない。

以上のように、3 自由度の力覚提示、提示力に慣性力や粘性力が含まれている、静力学世界に対してのみ有効であるなど手法毎に解決すべき課題がある。

## 3 研究の目的と提案手法の概要

本研究では計算量を分散させることが可能なマルチレートシステムにおいて 6 自由度の力覚提示の実現を

目指す。実現方法として、関連研究で紹介した中間表現と幾何ベースのレンダリングを組み合わせた力覚提示方法について提案する。

### 3.1 中間表現の拡張についての提案

シミュレーションベースの力覚レンダリングではバーチャルカップリングをさせる物体の慣性が提示力に含まれる。そのため、繊細な操作を必要とする状況などでは適さない。そこで、慣性力が提示力に含まれない幾何ベースの力覚レンダリングについて考える。Berkelman らの方法 [4] は各接触点の侵入量に対して、バネダンパモデルを利用した力覚レンダリングを行っていたため、提示力が不安定になっていた。つまり、中間表現を従来から扱われていた点と平面から点群と平面へと拡張し、6 自由度に対応させた拘束条件を立て、力覚ポイントが物体から侵入を解除するための力、トルクをレンダリングすることができれば、マルチレートシステムにおいて安定な 6 自由度力覚提示が実現できると考えられる。

他にも接触体積に基づいたペナルティ法のシミュレーション [9] を力覚レンダリングとして、マルチレートシステムに利用する方法が考えられるが、安定な力覚を提示するためには毎ステップ高速に体積の計算を行う必要がある。[9] によると 6 個の物体とインタラクションしている状態で衝突判定と接触判定にかかる計算時間は全計算時間の半分を占めている。そのため、上記の方法と比べて計算量が増加する可能性がある。

### 3.2 システム構成

図 2 に提案手法を実現するためのシステム構成図を示す。システムは物理シミュレーション、中間表現の作成を低速更新の物理スレッドで、力覚レンダリングを高速更新の力覚スレッドで行う。

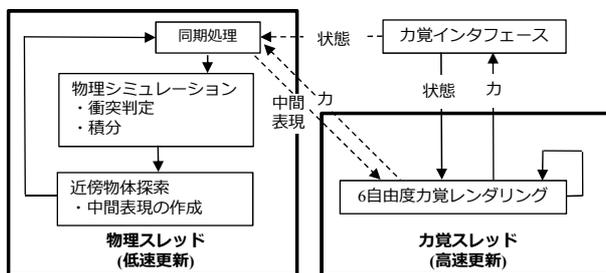


図 2 システム構成  
Fig.2 System Configuration

## 4 提案手法の実装

本章では提案する中間表現の作成方法、中間表現を利用した 6 自由度力覚レンダリングおよび摩擦力のレンダリングの実装方法について説明する。

### 4.1 6 自由度力覚レンダリングのための中間表現

本節では 6 自由度力覚レンダリングのための中間表現の作成方法について説明する。はじめに中間表現を作成するための準備について説明し、その後、中間表現の作成について説明する。中間表現を作成するための準備では [9] で提案されている接触解析の方法を利用する。

#### 4.1.1 力覚ポイント近傍の物体の探索

本研究では、ポリゴンで表現される複数の物体が存在するバーチャルな世界に対して、ポリゴンで表現される力覚ポイントをユーザが操作し、力覚インタラクションをすることを考える。マルチレートシステムでは力覚インタラクションの対象となる物体を限る（力覚ポイント近傍の物体を見つける）ことで力覚レンダリングの計算量を低減させることができる。力覚ポイント近傍の物体の探索は次のように行う。

まず、力覚ポイントについて、ある程度の大きさを持たせたバウンディングボックスを作り、*sweep and prune* [5] で大域接触判定を行う。そして、力覚ポイントとある程度絞られたバーチャル物体について GJK アルゴリズム [7] や Lin-Canny アルゴリズム [12], V-Clip アルゴリズム [17] 等を使い、最近傍点対を見つける (図 3 左上)。最後に最近傍点対の距離と任意に設定する閾値により、近傍物体を決定する。

#### 4.1.2 侵入領域の頂点の取得

次に力覚ポイントと前項で取得した近傍物体の侵入領域の頂点の取得方法について説明する。

侵入領域の頂点の取得は Muller ら [19] が提案する方法を用いる。まず、力覚ポイント近傍の物体を取得する際に得られる最近傍点対の中点を原点として、力覚ポイントと近傍物体の形状メッシュについてメッシュを点に変換する双対変換を行う。そして、双対変換された点群を含む最小の凸包を求め、凸包の面に対して双対変換を行うことで侵入領域の頂点を取得することができる (図 3 左下)。

#### 4.1.3 中間表現の作成

最後に中間表現の作成方法について説明する。まず、近傍物体は平面として表す。この平面は 4.1.1 項で得られる力覚ポイントと近傍物体の最近傍点対を結ぶベクトルを法線として、近傍物体側の最近傍点が載る平面とする。次に力覚ポイントは点群 (以下、接触点群と呼ぶ) として表す。接触点群は 4.1.2 項で得た侵入領域の各頂点をまとめたものである。ただし、近傍物体を表す平面上に載っている侵入領域の頂点は接触点群には含まない。そのため、本論文では接触点群として採用される侵入領域の頂点を接触点と呼ぶ。例えば、提案する中間表現は図 3 右のように表すことができる。ポリゴンで表される物体、力覚ポイントをそれぞれ平

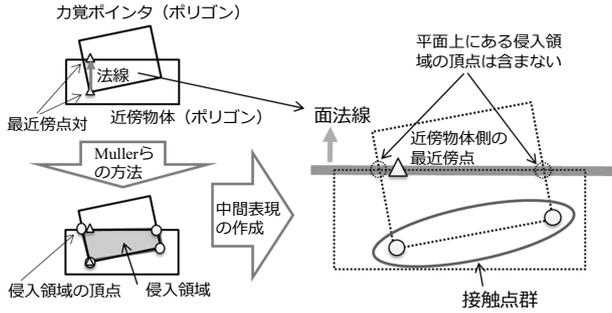


図3 6自由度力覚レンダリングのための中間表現

Fig.3 Intermediate representation for 6-DoF haptic rendering

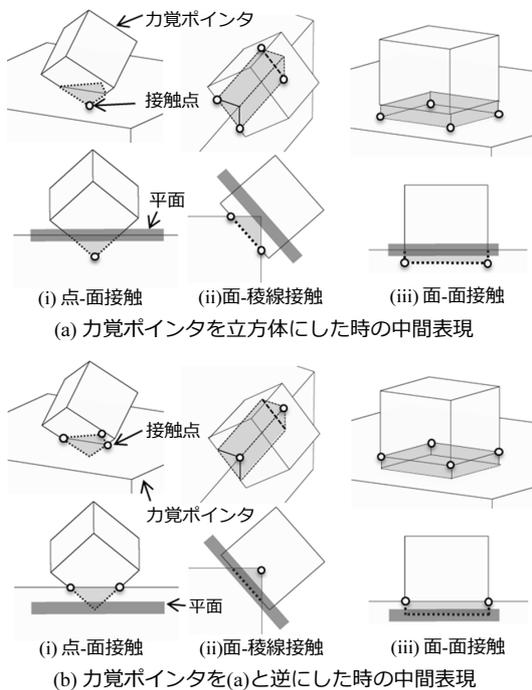


図4 中間表現の例

Fig.4 Examples of intermediate representation

面、接触点群とすることで、場合分けをせずに点、稜線、面の全ての組み合わせについて接触状態を表現することができる(図4)。

以上のように中間表現を作成し、6自由度力覚レンダリングを行う力覚スレッドへ送信する。送信する情報は下記となる。

- 平面を表す法線ベクトルと平面に載る点座標 (近傍物体側の最近傍点)
- 接触点群の座標 (平面上に載らない力覚ポインタの侵入領域の各頂点)

#### 4.2 6自由度力覚レンダリング

力覚レンダリングは横山ら[26]が提案する拘束ベースの侵入解除を参考にした。これは物体同士の侵入

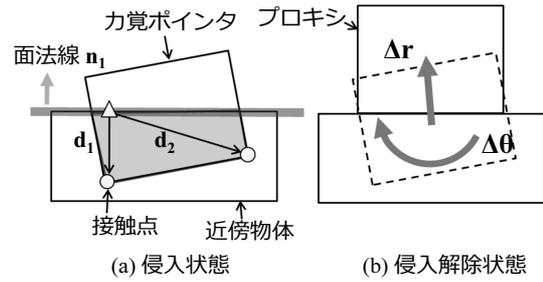


図5 力覚ポインタの侵入解除

Fig.5 Releasing penetration of haptic pointer

について幾何学的な拘束を立て、それを解くことで侵入解除状態を求める方法である。横山らは物理シミュレーションの衝突応答に拘束ベースの侵入解除を利用し、バーチャル世界内でユーザーが操作する物体と力覚ポインタをバーチャルカップリングさせることで力覚提示を実現している。操作物体が無拘束の時には提示力を出力しないようにしているため、操作物体自身の慣性力は提示されないが、操作物体を床面と接触させスライドさせると慣性力が提示されてしまう。

我々は力覚ポインタと近傍物体が侵入した際に、拘束ベースの侵入解除を用いて力覚ポインタが物体に侵入しない状態となる移動量(以下、侵入解除量と呼ぶ)を計算する。そして、侵入解除量にバネダンパ係数を乗じ、提示力を計算する。侵入解除量を直接利用することで、慣性力が含まれない力覚提示が可能となる。

本項では力覚スレッド1ステップ間で行う力覚レンダリングの方法について説明する。

図5.(a)のように力覚ポインタが近傍物体に侵入している状態を考える。まず、平面の法線を  $\mathbf{n}_i \in \mathbb{R}^3$  とし、4.1.3項で得た近傍物体側の最近傍点から接触点へのベクトルを  $\mathbf{d}_i \in \mathbb{R}^3$  とする。 $i$ は力覚ポインタが侵入している  $i$ 番目の接触点である。ベクトル  $\mathbf{d}_i$  と平面の法線  $\mathbf{n}_i$  との内積をとり、 $\mathbf{d}_i \cdot \mathbf{n}_i < 0$ であれば接触点は平面(近傍物体)に侵入していることになる。そこで、近傍物体に侵入している接触点が侵入解除、つまり内積が  $\mathbf{d}_i \cdot \mathbf{n}_i \geq 0$ となるように力覚ポインタの中心の並進量、回転量を計算する。このとき、力覚ポインタの位置姿勢に並進量、回転量を足し合わせたものがプロキシ(力覚ポインタの侵入解除状態)となる。接触点の侵入を解除するための条件式は

$$\Delta\theta \times \mathbf{r}_i + \Delta\mathbf{r} \geq \|\mathbf{d}_i \cdot \mathbf{n}_i\| \mathbf{n}_i \quad (1)$$

と書ける。ここで、 $\mathbf{r}_i \in \mathbb{R}^3$ は力覚ポインタの中心から接触点までのベクトル、 $\Delta\mathbf{r} \in \mathbb{R}^3$ は力覚ポインタの侵入を解除するための並進量、 $\Delta\theta \in \mathbb{R}^3$ は力覚ポインタの侵入を解除するための回転量である。並進量、回

転量はそれぞれ

$$\Delta \mathbf{r} = \sum_i^n f_i \mathbf{n}_i \quad (2)$$

$$\Delta \boldsymbol{\theta} = \sum_i^n f_i \mathbf{R}_w^{-1} (\mathbf{r}_i \times \mathbf{n}_i) \quad (3)$$

と表せる. この時,

$\mathbf{R}_w \in \mathbb{R}^{3 \times 3}$ : 力覚ポイントの並進移動に対する  
回転移動の重み行列

$f_i (> 0)$ : 力覚ポイントの接触点が並進移動  
に作用する大きさ

$n$ : 接触点の数

である.

次に, 式 (2), 式 (3) を式 (1) に代入し, 接触点数  $n$   
本分の式をまとめると

$$\mathbf{C} \mathbf{f} \geq \mathbf{d} \quad (4)$$

となる. ただし,  $\mathbf{f} = (f_1 \ f_2 \ \dots \ f_n)^T$ ,

$\mathbf{d} = (\|d_1\| \ \|d_2\| \ \dots \ \|d_n\|)^T$  で行列  $\mathbf{C} \in \mathbb{R}^{n \times n}$  の要素  $c_{ij}$   
は

$$c_{ij} = (\mathbf{r}_i \times \mathbf{n}_i)^T \mathbf{R}_w^{-1} (\mathbf{r}_j \times \mathbf{n}_j) + \mathbf{n}_i^T \cdot \mathbf{n}_j \quad (5)$$

である. 連立 1 次不等式である式 (4) の  $\mathbf{f}$  を求め, 式  
(2), 式 (3) に代入することで, 侵入解除量を求めること  
ができる. 今回の実装では実時間性を確保するために,  
計算を規定内に打ち切ることができるガウスザイ  
デル法を利用した.

ガウスザイデル法は連立 1 次方程式を解くための解  
法であるが, 解きたい変数を更新する度に, 変数が条  
件を満たしているかを判定し, 満たしていない場合は  
変数の条件を満たすように拘束条件を入れることで不  
等式を解くことができる. 式では次のように表すこと  
ができる.

$$f_i^{(k+1)} = \frac{1}{c_{ii}} \left( d_i - \sum_{j=1}^{i-1} c_{ij} f_j^{(k+1)} - \sum_{j=i+1}^n c_{ij} f_j^{(k)} \right) \quad (6)$$

$$\text{if } f_i^{(k+1)} < 0 \text{ then } f_i^{(k+1)} = 0$$

ここで  $f$  の上付き文字は反復回数で,  $f_i^{(k+1)}$  は  $f_i$  の  $k$   
回目の更新である.  $\mathbf{f}$  が収束するように反復回数を設  
定し, 式 (6) に従って  $\mathbf{f}$  を求める.

最後に提示力, 提示トルクの計算について説明する.  
提示力  $\mathbf{f}_d$ , 提示トルク  $\boldsymbol{\tau}_d$  の計算は侵入解除量  $\Delta \mathbf{r}$ ,  $\Delta \boldsymbol{\theta}$   
を用いて,

$$\mathbf{f}_d(t) = k_t \Delta \mathbf{r}(t) + d_t \frac{\Delta \mathbf{r}(t) - \Delta \mathbf{r}(t - \Delta t^L)}{\Delta t^L} \quad (7)$$

$$\boldsymbol{\tau}_d(t) = k_r \Delta \boldsymbol{\theta}(t) + d_r \frac{\Delta \boldsymbol{\theta}(t) - \Delta \boldsymbol{\theta}(t - \Delta t^L)}{\Delta t^L} \quad (8)$$

となる. ただし,  $k_t$ ,  $k_r$  は並進, 回転のバネ係数,  $d_t$ ,  
 $d_r$  は並進, 回転のダンパ係数,  $\Delta t^L$  は力覚スレドの  
更新時間である.

本項では力覚スレド 1 ステップの間の力覚レンダ  
リングについて説明した. 実際は物理スレド 1 ス  
テップに対して複数回力覚スレドのステップ繰り返  
す. その間の力覚ポイントと近傍物体の接触状態は線  
形的に変化すると仮定し, 平面と接触点群はそれぞれ  
補間をさせながら力覚レンダリングを行う.

### 4.3 摩擦力のレンダリング

前述の抗力のレンダリングは力覚ポイントが物体に  
侵入している状態から法線方向へ侵入が解除されてい  
る状態, つまり, プロキシの位置姿勢を決定すること  
で実現している. 摩擦力のレンダリングは, まずクー  
ロンの摩擦モデルに基づいた摩擦円錐 [22][27] を用い  
て, 法線の垂直方向に対するプロキシの位置姿勢を求  
める. そして, 力覚ポイントとプロキシの距離からバ  
ネダンパモデルを用いて提示力を計算する. 本節では  
摩擦力のレンダリングの方法について説明する.

まず, 図 6.(a) のように力覚ポイントが物体に接触し  
ている状態から水平方向に移動させる場合を考える.  
この時の力覚ポイントの位置姿勢を  $\mathbf{p} \in \mathbb{R}^6$ , プロキシ  
の位置姿勢を  $\mathbf{q} \in \mathbb{R}^6$  とする. 時刻  $t-1$  の時, 力覚ポ  
イントが物体に侵入し  $\mathbf{p}_{(t-1)}$  の状態になるとする. す  
ると, 4.2 節の方法によりプロキシの位置姿勢  $\mathbf{q}_{(t-1)}$  が  
決まる. この状態から時刻  $t$  に, 力覚ポイントが  $\mathbf{p}_{(t)}$   
へ移動すると中間表現の作成 (4.1 節) によって, 新  
たに接触点  $\mathbf{v}_{pi} \in \mathbb{R}^3$  が得られる. そして接触点  $\mathbf{v}_{pi}$  を  
平面に射影したものを  $\mathbf{v}_{psi}$ , プロキシ位置姿勢  $\mathbf{q}_{(t-1)}$   
に対して射影したものを  $\mathbf{v}_{qsi}$  とする.

次に図 6.(b) のように力覚ポイントの各接触点に対  
して摩擦係数  $\mu$  の摩擦円錐を張る. そして, 点  $\mathbf{v}_{psi}$  か  
らベクトル  $\mathbf{v}_{qsi} - \mathbf{v}_{psi}$  と摩擦円錐が交差する点までの  
ベクトル  $\mathbf{l}_i$  を求める. ベクトル  $\mathbf{l}_i$  を利用することで平  
面に対して水平方向の力覚ポイントの拘束条件が求め  
られ, 摩擦状態を切り替えることができる (図 6.(c)).  
摩擦状態の切り替わりは次のように表せる.

$$\mathbf{v}_{qni} = \begin{cases} \mathbf{v}_{qsi} & \text{if } \|\mathbf{v}_{qsi} - \mathbf{v}_{psi}\| \leq \|\mathbf{l}_i\| \\ \Rightarrow \text{静止摩擦} \\ \mathbf{v}_{psi} + \mathbf{l}_i & \text{if } \|\mathbf{v}_{qsi} - \mathbf{v}_{psi}\| > \|\mathbf{l}_i\| \\ \Rightarrow \text{動摩擦} \end{cases} \quad (9)$$

以上の条件を利用すると, プロキシの水平方向の移動  
を決定するための条件式は

$$\Delta \boldsymbol{\theta} \times \mathbf{r}_i + \Delta \mathbf{r} \geq \mathbf{v}_{qni} - \mathbf{v}_{psi} \quad (10)$$

となり, 式 (1) と同様の形で表せる. つまり, 抗力と  
摩擦力のレンダリングのためのプロキシの位置姿勢計  
算は式 (4) の形にまとめることができる.

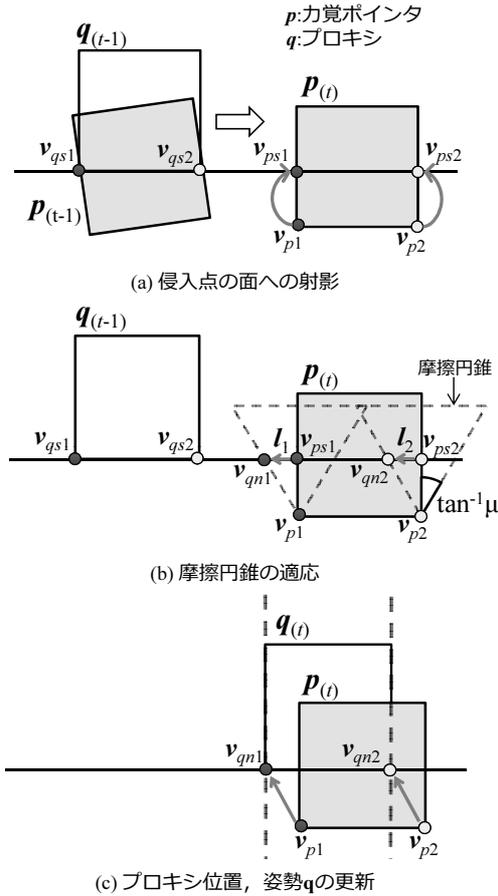


図 6 摩擦力提示計算の手順  
 Fig. 6 Procedure of calculating friction force for haptic display

#### 4.4 提案手法の計算手順

抗力と摩擦力の計算をまとめると、次の通りになる。  
 物理スレッド

1. 平面と接触点群で表される中間表現を作成する。
2. 力覚スレッドへ中間表現を送信する。

力覚スレッド

1. 中間表現を受信する。
2. 式 (1) を使い、法線方向のプロキシの位置姿勢を決定するための条件を接触点  $n$  本分たてる。
3. 各接触点に対して、摩擦円錐を張る。現在の力覚ポイントの位置姿勢、前回のプロキシの位置姿勢から式 (9) に基づいて、各接触点の摩擦状態を決める。
4. 式 (10) を使い、法線に対して水平方向のプロキシの位置姿勢を決定するための条件を接触点  $n$  本分たてる。
5. 手順 2, 4 で求めた拘束条件を式 (4) の形にまとめ、 $f$  を求める。そして式 (2), 式 (3) から力覚ポイントの移動量  $\Delta r$ ,  $\Delta \theta$  を求める。
6. 力覚ポイントの移動量  $\Delta r$ ,  $\Delta \theta$  を使い、提示力、

トルクを式 (7), (8) から計算する。

## 5 評価

提案手法の有効性を評価するために、6 自由度の力覚提示、摩擦力の提示、計算時間を調べた。

### 5.1 実験環境

実験環境は計算機 Windows7 x86 (Intel(R) Core(TM) 2 Duo CPU 2.33GHz), 力覚インタフェース SPIDAR-G6 [11], 物理シミュレータ Springhead2 [10] で構成した。システムの更新周期は物理スレッドを 20ms, 力覚スレッドを 1ms と設定した。

### 5.2 6 自由度の力覚提示の評価

提案手法が適切に 6 自由度の力覚提示を行えることを確認するために実験を行った。実験は図 7 のように力覚ポイント进行操作する。手順は以下の通りである。

- (a) 力覚ポイントが 45 度回転させた状態から y 軸負方向に移動させ、稜線と床面を接触させる。
- (b) 力覚ポイントの稜線と床面を接触させた状態で、力覚ポイントが z 軸周りに負方向に回転させる。
- (c) 力覚ポイントの面を床面に接触させる。

実験中に力覚ポイントの位置姿勢と力覚ポイントが受ける力、トルクを計測し、力、トルクの計算が妥当であるか確認する。実験条件として、力覚ポイントの形状は 1 辺 0.2m の立方体で回転の重み行列を  $R_{wxx} = R_{wyy} = R_{wzz} = 3.33$ , 力覚レンダリングのためのバネダンパ係数を並進は  $k_t = 9000 \text{ N/m}$ ,  $d_t = 0 \text{ Ns/m}$ , 回転は  $k_r = 9000 \text{ Nm/rad}$ ,  $d_r = 0 \text{ Nms/rad}$ , 力覚ポイントと床との間の摩擦係数を  $\mu = 0$  とした。

実験結果を図 8, 9 に示す。力覚ポイントについて、 $y$  は y 軸方向の中心位置、 $\theta_z$  は中心 z 軸周りの回転角度、 $f_y$  は y 軸方向に加わる力、 $\tau_z$  は z 軸周りに加わるトルクを表す。図 8, 9 の (a), (b), (c) は上記の手順 (a), (b), (c) を行っている期間に相当する。

まず、期間 (b) で力覚ポイントの稜線と床面が接触し、y 軸正方向に力が発生している。接触した瞬間は力の方向に力覚ポイントの中心位置があるため、z 軸周りのトルクは発生していないが、力覚ポイントを徐々に回転させることで、z 軸周りに負方向のトルクが増加していることがわかる。そして、力覚ポイントの傾きが 0 度に近づくとトルクの大きさは最大となり、傾きが 0 度になった時点で急激にトルクが小さくなっている。

このことから提示トルク  $\tau_z$  の計算は力覚ポイントの傾きに応じて正しく計算されていることがわかる。ただし、図 8 の 3.8s 付近で y 軸方向の提示力  $f_y$  が 3N から 2N へ一旦低下し、その後 3.5N に増加していることが確認できる。ここでは、力覚ポイントの接触が 2 点の接触点 (稜線接触) から 4 点の接触点 (面接触)

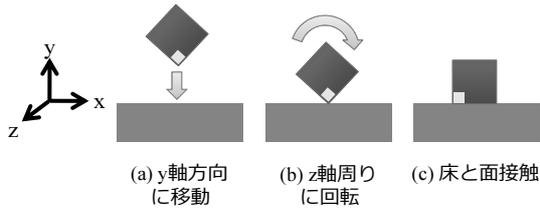


図 7 実験中の操作  
Fig. 7 Manipulation in the experiment



図 10 摩擦力提示実験  
Fig. 10 Experiment of displaying friction force

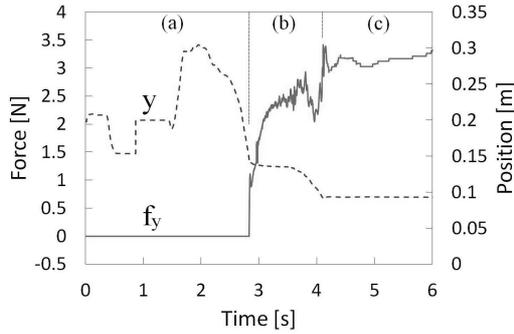


図 8 力覚ポインタ y 方向の位置, 加わった力  
Fig. 8 Applied force and the position of the haptic pointer (y-axis)

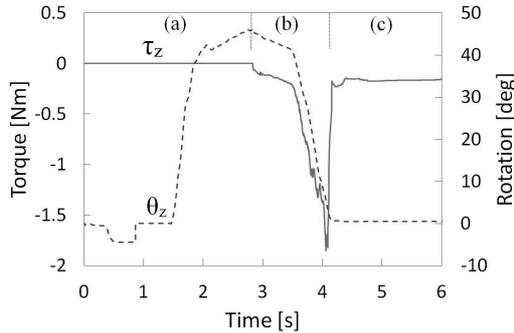


図 9 力覚ポインタ z 軸周りの角度, 加わったトルク  
Fig. 9 Applied torque and the rotation of the haptic pointer (around z-axis)

へ向かう状態である。この接触状態の遷移過程では、まず力覚ポインタと床面の侵入が浅くなり、提示力が低下する。そして、面接触になることで侵入が深くなり、提示力が増加したと考えられる。

また、(c) の力覚ポインタが床面に面接触している際にも  $-0.2\text{Nm}$  程度のトルクが発生している。これは、力覚ポインタが床面に対して平行に接触していないために起きている。力覚ポインタの姿勢はユーザーの操作によって決定されるが、ユーザーは力覚ポインタを正確に操作することは難しいため、若干のトルクが発生する。

### 5.3 摩擦力提示の評価

提案手法が適切に摩擦力の提示を行えることを確認するための実験を行った。実験は図 10 のように直方体の形状をした力覚ポインタをバーチャル物体に接触させ、バーチャル物体に貼られているテクスチャをなぞる。その間に力覚ポインタに加わる抗力、摩擦力を計測する。

実験では力覚ポインタについて、回転の重み行列を  $R_{wxx} = R_{wzz} = 8420$ ,  $R_{wyy} = 1670$ , 力覚レンダリングのためのバネダンパ係数を並進は  $k_t = 9000 \text{ N/m}$ ,  $d_t = 10 \text{ Ns/m}$ , 回転は  $k_r = 300 \text{ Nm/rad}$ ,  $d_r = 0 \text{ Nms/rad}$ , 力覚ポインタとバーチャル物体との間の摩擦係数を  $\mu = 0.5$  とした。

実験結果を図 11, 12 に示す。図 11 は力覚ポインタに加わった力の方向を  $100\text{ms}$  のサンプリングでプロットしたもので、図 12 は力覚ポインタに加わった力で抗力 (y 軸方向), 摩擦力 (xz 平面に平行な力), 回転トルク (y 軸周りのトルク) をそれぞれ  $1\text{ms}$  のサンプリングでプロットしたものである。図 11 より、なぞり操作時に力覚ポインタに加わる力は力覚ポインタの移動に対して逆向きに働いていることがわかる。このことから適切な向きに摩擦力が計算されているといえる。

次に図 12 を見ると、抗力と摩擦力の波形が一致していない部分 (楕円で囲まれた部分 (a), (b), (c)) が存在する。その他の部分では摩擦力に摩擦係数  $\mu=0.5$  を除すると、抗力と一致することから楕円内の部分で動摩擦状態と静止摩擦状態が切り替わっていることがわかる。この部分は図 11 のなぞり操作開始時 (a), 力覚ポインタの移動の切り返し (b), なぞり操作終了時 (c) に相当し、摩擦状態が適切に遷移しているといえる。また、回転トルクが発生していることから回転摩擦力が生じていることが伺え、6 自由度の力覚レンダリングが行えていることがわかる。

### 5.4 シミュレーションによる計算時間の評価

提案手法では物理シミュレーションを低速更新の物理スレッド、力覚レンダリングを高速更新の力覚スレッドで実行させることで計算量の分散を図っている。そこで、従来手法と比較して提案手法の計算時間の評価

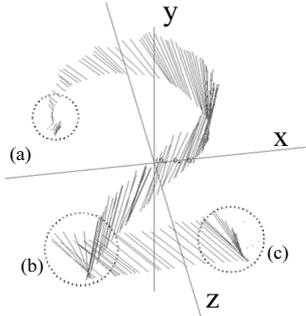


図 11 なぞり操作時の提示力の方向  
Fig. 11 Direction of the force during tracing

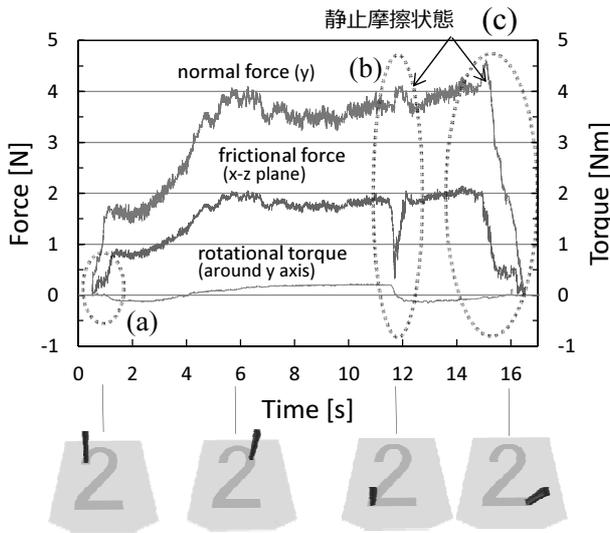


図 12 抗力, 摩擦力, 回転トルクの大きさ  
Fig. 12 Norm of normal force, friction force and rotational torque

を行った。評価の方法として、図 13 に示すように直立方体の形状を持つ力覚ポイントで、1 列 3 個の立方体を押している間のシミュレーション 1 ステップ当たりの計算時間を計測した。立方体の数は 1 列 3 個ずつ増やしていき、それぞれについて計算時間を測定した。比較した手法は、提案手法 (物理スレッド 20ms 更新, 力覚スレッド 1ms 更新), ペナルティ法 [9](1ms 更新), バーチャルカップリングによる方法 [6] (1ms 更新) である。時間の計測には Windows の標準 API で  $\mu s$  単位で計測可能な QueryPerformanceTimer を利用した。

実験結果を図 14 に示す。図の横軸は立方体の個数、縦軸は実時間内計算率 (測定した計算時間を更新時間で除した値) である。提案手法はマルチレートシステム上で実装されているため、力覚レンダリングにかかった計算時間を力覚スレッドの更新時間で除した値と物理シミュレーションにかかった計算時間を物理スレッドの更新時間で除した値の和をプロットしてある。図 14 を見ると、全ての手法について立方体の増加に従い線形的に計算時間が増加している。ペナルティ法

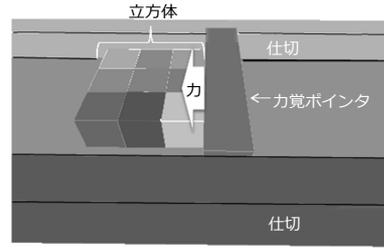


図 13 計算時間評価のためのシミュレーション構成  
Fig. 13 Configuration of the simulation speed test

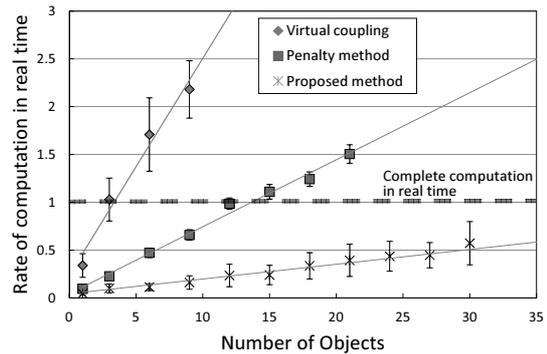


図 14 力覚提示システムの実時間内計算率  
Fig. 14 Rate of computation in real time

では立方体数が約 12 個以上、バーチャルカップリングによる方法では約 3 個以上で実時間内計算率が 1 を越え、実時間で計算が終了していない。一方、提案手法では立方体が約 30 個の場合でも実時間内計算率が 1 以下であり、実時間で計算が終了している。図には記載していないが、提案手法は約 60 個まで実時間で計算可能であった。以上をまとめると、提案手法は計算量を分散させることで、ペナルティ法と比較して約 5 倍、バーチャルカップリングによる方法と比較して約 20 倍の立方体を実時間で計算可能であり、計算量を必要とするバーチャル世界に対しての力覚インタラクションに適しているといえる。

### 5.5 力覚ポイントの接触点数毎の計算時間の評価

提案手法では力覚ポイントの侵入解除状態を求めするために接触点の数だけ拘束を解く必要がある。つまり、接触点の数が増加すると、拘束を解く数も増加し、計算量が増加する。そこで、提案手法はどの程度の接触点数まで実時間で計算が可能であるかをシミュレーションによって調べた。実験内容は力覚ポイントの形状を多角柱とし、バーチャルな床面と接触させ、横方向にスライドさせた時の力覚レンダリングの計算時間を測定した (図 15 左)。測定は摩擦なしの場合と摩擦ありの場合の 2 通りを 1ms の更新時間で行った。

シミュレーション結果を図 15 右に示す。横軸は接

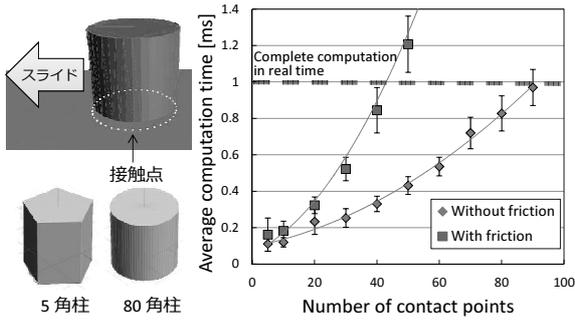


図 15 接触点数毎の計算時間

Fig. 15 Computation time by the number of contact points

触点の数, 縦軸は計算時間である. 実時間で計算可能であった接触点の数は摩擦なしの場合は約 90 点, 摩擦ありの場合は約 40 点であった. 摩擦なしの場合と摩擦ありの場合で計算時間が異なるのは, 摩擦ありの場合のほうが拘束を解く数が 2 倍となるためである. 一般にガウスザイデル法を利用して密な行列を解く場合の計算量は  $O(n^2)$  である. 提案手法で解く行列は密な行列であるため, 今回のシミュレーション結果も曲線を描いている. 以上から, 提案手法は数十程度の接触点であれば実時間で利用できるといえる.

## 5.6 考察

### 5.6.1 プロキシ位置姿勢計算のための回転の重み行列

提案手法はプロキシ位置姿勢を計算する過程で回転の重み行列  $R_w$  を導入している. 力覚ポインタの侵入解除は重み行列の大きさにより, 並進移動量と回転移動量の割合が決まる. 重み行列の値を適切に設定しないと回転移動量が小さくなり回転トルクの力覚提示が小さくなること (図 16.(a)) や回転移動量が大きくなりプロキシが物体に侵入してしまうこと (図 16.(b) の☆の部分) が起きる. 特に後者の場合, 力覚ポインタを点や稜線接触させた状態から面接触に遷移させると, 侵入解除のための並進移動量と回転移動量がそれぞれ大きく変化し, 提示力, 提示トルクが安定しない.

この問題を解消するためには, 力覚ポインタについて侵入の可能性がある頂点を全て列挙し, その頂点が侵入しないための拘束を侵入解除計算に加える必要がある. 全ての頂点を列挙するためには近傍点を求める段階で力覚ポインタを適当に回転させ, 侵入の可能性がある頂点を探索すれば良いが計算量や効率性を考えると現実的ではない.

本研究では対処方法として回転の重み行列の値を大きく設定し, 回転移動量を小さくする. そして, 回転移動量を小さくする分だけ力覚レンダリングの回転のバネダンパ係数を大きくすることでプロキシが侵入す

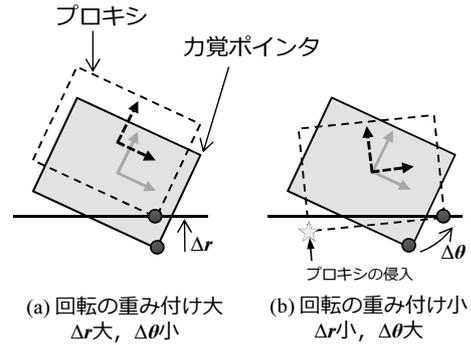


図 16 回転の重み行列による侵入解除状態の変化

Fig. 16 Changes of the proxy state by the weighting matrix of rotation

る問題を低減させている. 今回行った評価実験では, まず力覚ポインタの質量を 1 と仮定したときの慣性テンソルを求め回転の重み行列とし, 安定に力覚が提示できる範囲でバネダンパ係数を調整した. しかし, 上記のプロキシが物体に侵入する問題が起きるので, プロキシが大きく回転ないように回転の重み行列に倍数をかけ, その倍数分だけバネダンパ係数を小さくした.

## 6 まとめと今後の展望

本論文ではマルチレートな剛体力覚提示システムのための 6 自由度力覚提示手法について提案した. 提案手法では計算量を分散させるためのマルチレートシステム, 力覚ポインタ自身の慣性が含まれない 6 自由度力覚提示を実現するために力覚レンダリングに必要な情報を平面と接触点群で表される中間表現としてまとめ, 更新周期の異なるスレッド間で送受信させる. そして, 中間表現を利用した幾何学的な拘束条件からプロキシ位置姿勢を算出し力覚レンダリングに利用することで, 6 自由度の力覚提示を実現した. また, 摩擦円錐を利用した摩擦力の提示計算を導入することで, 静動切り替わりのある摩擦力の提示を実現した.

最後に, 提案手法を実装したシステムについて実験を行い, 6 自由度の力覚提示, 摩擦力の提示が適切に行っていることを確認した. また, 提案手法と従来手法について計算時間を比較し, 提案手法は従来手法より多くの物体を実時間でシミュレーション可能であることがわかった.

今後の展望として応答性の向上が挙げられる. マルチレートシステムは定期的に各スレッドを同期させる必要があり, 同期のために時間遅れが発生する. 今回, 実装したシステムでは力覚インタラクションの対象となる物体に加わる力 (提示力の反作用) を高速更新させる力覚スレッド内で計算する. そのため, 物体に加

わる力が物理シミュレーションに反映されるまでに物理スレッドの更新1ステップ分だけ遅れることになり、応答性が必要な操作には適していない。例えば筆者らが提案する局所シミュレーション [25] を利用したシステムと組み合わせることなどが考えられる。

### 参考文献

- [1] Y. Adachi, T. Kumano, and K. Ogino. Intermediate representation for stiff virtual objects. *Virtual Reality Annual International Symposium*, pp. 203–210, March 1995.
- [2] K. Akahane, S. Hasegawa, Y. Koike, and M. Sato. A proposal of a high definition haptic rendering for stability and fidelity. *ICAT2006*, pp. 162–167, November 2006.
- [3] D. Baraff. Fast contact force computation for nonpenetrating rigid bodies, 1994.
- [4] P. J. Berkelman and R. L. Hollis. Lorentz magnetic levitation for haptic interaction: Device design, performance, and integration with physical simulations. *The International Journal of Robotics Research*, 19(7):644–667, 2000.
- [5] J. D. Cohen, M. C. Lin, D. Manocha, and M. Ponamgi. I-collide: an interactive and exact collision detection system for large-scale environments. *Proc. of the 1995 symposium on Interactive 3D graphics*, 1995.
- [6] J. E. Colgate, M. C. Stanley, and J. Michael. Issues in the haptic display of tool use. *IEEE/RSJ International Conference on Intelligent*, pp. 140–145, 1995.
- [7] E. Gilbert, D. Johnson, and S. S. Keerthi. A fast procedure for computing the distance between complex objects in three-dimensional space. *IEEE Journal of Robotics and Automation*, 4(2):193–203, 1988.
- [8] S. Hasegawa, M. Ishii, Y. Koike, and M. Sato. Inter-process communication for force display of dynamic virtual world. *Proc. of the ASME-Dynamic Systems and Control Division-1999*, 67:211–218, 1999.
- [9] S. Hasegawa and M. Sato. Real-time rigid body simulation for haptic interactions based on contact volume of polygonal objects. *Computer Graphics Forum*, 23(3):529–538, 2004.
- [10] S. Hasegawa and Springhead Development Team. Springhead2. <http://springhead.info/wiki/>.
- [11] S. Kim, J. Berkley, and M. Sato. A novel seven degree of freedom haptic device for engineering design. *VIRTUAL REALITY*, 6(4):217–228, 2003.
- [12] M. C. Lin and J. F. Canny. A fast algorithm for incremental distance calculation. *In IEEE International Conference on Robotics and Automation*, pp. 1008–1014, 1991.
- [13] L. Love and W. Book. Contact stability analysis of virtual walls. *Proc. Of Dynamic Systems and Control Division ASM*, pp. 689–694, 1995.
- [14] W. R. Mark, S. C. Randolph, M. Finch, J. M. V. Verth, and R. M. Taylor II. Adding force feedback to graphics systems: Issues and solutions. *Proc. of the 23rd annual conference on Computer graphics and interactive techniques*, pp. 447–452, 1996.
- [15] W. A. Mcneely, K. D. Puterbaugh, and J. J. Troy. Six degree-of-freedom haptic rendering using voxel sampling. *In Proc. of ACM SIGGRAPH*, pp. 401–408, 1999.
- [16] C. A. Mendoza and C. Laugier. Realistic haptic rendering for highly deformable virtual objects. *In Proceedings of the Virtual Reality 2001 Conference (VR'01)*, VR '01, pp. 264–269, Washington, DC, USA, 2001. IEEE Computer Society.
- [17] B. Mirtich. V-clip: fast and robust polyhedral collision detection. *ACM Transactions on Graphics*, 17:177–208, 1998.
- [18] H. Mitake, S. Hasegawa, and M. Sato. Reactive virtual creatures for dexterous physical interactions. *Motion in Games 2011*, 2011.
- [19] D. E. Muller and F. P. Preparata. Finding the intersection of two convex. *Theoretical Computer Science*, 7(2):217–236, 1978.
- [20] M. Ortega, S. Redon, and S. Coquillart. A six degree-of-freedom god-object method for haptic display of rigid-bodies. *Proc. of IEEE Virtual Reality Conf.*, 2006.
- [21] M. A. Otaduy and M. C. Lin. A modular haptic rendering algorithm for stable and transparent 6-dof manipulation. *IEEE Transactions on Robotics*, 22(4):751–762, August 2006.
- [22] D. Ruspini, K. Kolarov, and O. Khatib. The haptic display of complex graphical environments. *SIGGRAPH 97 Proceedings*, 1997.
- [23] R. Weller and G. Zachmann. Inner sphere trees for proximity and penetration queries. *Robotics Science and Systems Conference (RSS)*, 2009.
- [24] C. B. Zilles and J. K. Salisbury. A constraint-based god object method for haptics display. *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robotics and Systems*, 1995.
- [25] 須佐, 大内, 岩下, 佐藤, 長谷川. 局所的な高速物理シミュレーションによる高解像度力覚レンダリングの実現. *日本バーチャルリアリティ学会論文誌*, 14(4):463–471, 2009.
- [26] 横山, 飯田. デジタルモックアップのための剛体運動シミュレーションに関する研究. *岐阜県生産情報技術研究所研究報告*, 4:51–56, 2002.
- [27] 菊植, 藤本. 幾何学的力覚提示アルゴリズムの力学的解釈とインピーダンス型およびアドミッタンス型の実装法. *日本ロボット学会誌*, 25(2):142–151, 3 2007.

(2011年11月7日受付)

### [著者紹介]

#### 須佐 育弥 (学生会員)



2008年電気通信大学電気通信学部知能機械工学科卒業, 2010年同大学大学院電気通信学研究科知能機械工学専攻修士課程修了, 同年東京工業大学大学院総合理工学研究科知能システム科学専攻博士後期課程在学中, 2011年日本学術振興会特別研究員. バーチャルリアリティ, 力覚インタラクシオンの研究に従事.

#### 長谷川 晶一 (正会員)



1997年東京工業大学工学部電気電子工学科卒業, 1999年同大学大学院知能システム科学専攻修士課程修了, 同年ソニー株式会社入社, 2000年東京工業大学精密工学研究所助手, 2007年電気通信大学知能機械工学科准教授, 2010年東京工業大学精密工学研究所准教授, 現在に至る. バーチャルリアリティ, 力覚インタフェース, ヒューマンインタフェースの研究に従事. 工学博士