

論文 / 著書情報
Article / Book Information

題目(和文)	高品質な力覚インタラクションのためのマルチレートシステム
Title(English)	
著者(和文)	須佐育弥
Author(English)	Ikumi Susa
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第8957号, 授与年月日:2012年9月25日, 学位の種別:課程博士, 審査員:長谷川 晶一
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第8957号, Conferred date:2012/9/25, Degree Type:Course doctor, Examiner:
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

博士論文

高品質な力覚インタラクションのための
マルチレートシステム

須佐 育弥

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

2012 年 9 月

高品質な力覚インタラクションのための マルチレートシステム

博士論文審査委員会

主査	長谷川 晶一	准教授
委員	佐藤 誠	教授
委員	長橋 宏	教授
委員	石井 秀明	准教授
委員	小野 功	准教授

著作權所有者

須 佐 育 弥

2012

高品質な力覚インタラクションのための マルチレートシステム

須佐 育弥

概要

本論文では「高品質な力覚インタラクションのためのマルチレートシステム」と題し、6自由度の力覚提示に必要な情報と動力学の力覚提示に必要な情報を明らかにし、それぞれの情報を利用した力覚提示システムの提案をした。そして、高品質な力覚インタラクションを実現するマルチレート力覚提示システムを実現した。本論文は5章から構成される。

第1章「序論」ではバーチャル世界との力覚インタラクションについて述べ、力覚インタラクションが設計、教育、訓練、エンタテインメント等の分野において応用されていることを述べた。また、力覚提示システム、特に剛体のバーチャル世界との力覚インタラクションを実現するために必要な要素である物理シミュレーション、力覚インタフェース、力覚レンダリングを挙げ、それぞれについて説明を行った。そして、これらの要素により構成される従来の力覚提示システムについて触れ、特徴、問題点を述べ、高品質な力覚インタラクションのための力覚提示システムの実現することを本研究の目的とした。目的の実現のために、6自由度の力覚提示に必要な情報と動力学の力覚提示に必要な情報を明らかにし、それぞれの情報を利用した力覚提示システムの提案をすることを述べた。

第2章「マルチレート力覚提示システムと課題」では大規模なバーチャル世界との力覚インタラクションを実現する方法として従来から提案されているマルチレートシステムの研究を紹介した。さらに、従来から提案されている手法について問題点と原因を考察し、高品質な力覚インタラクションを実現するために必要な事項を検討した。そして、不要な慣性力のでない6自由度の力覚提示、規模の大きいバーチャル世界、時間遅れのない応答、整合性が課題であると述べた。

第3章「6自由度力覚レンダリングのための中間表現」では6自由度力覚レンダリングに必要な情報である幾何学的な中間表現を明らかにし、マルチレートな6自由度力覚提示システムを提案した。6自由度の力覚レンダリングに必要な情報を明らかにすることで、大規模なバーチャル世界に対して不要な慣性力が入らない6自由度の力覚インタラクションを実現した。6自由度の力覚レンダリングに必要な情報は、力覚ポインタ（ユーザがバーチャル世界内で操作する物体）がバーチャル物体と接触した時の、力覚ポインタの接触点群とバーチャル物体に載る平面（力覚ポインタとバーチャル物体間の抗力方向を法線とする平面）である。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評価を行い、6自由度の力覚提示、摩擦力の提示が適切に行え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確認できた。

第4章「局所シミュレーションを用いた力覚提示システム」では動力学の力覚提示に必要な情報を明らかにし、ユーザの入力に対してシステムの応答に遅れのない正確な力覚提示が可能なシステムを実現した。動力学の力覚提示に必要な情報とは力と加速度の比例係数であるアクセラランスと加速度項で、動力的な中間表現といえる。アクセラランスを利用すると、力覚ポインタ近傍の物体の運動を高速にシミュレート（局所シミュレーション）することができ、ユーザが加える力を即座に反映させることができる。さらに、近傍物体以外の物体の運動は低速にシミュレート（大域シミュレーション）されるので、計算負荷を抑えることができる。評価としてシミュレーションを行い、提案手法が従来マルチレートシステムに比べて正確な力覚レンダリングが行えていることを確認した。また、各手法の計算時間を測定し、シミュレータを高速更新する手法に比べて提案手法は大規模なバーチャル世界上で力覚提示が可能であることを確認した。最後に本手法の適用範囲は3自由度の力覚インタラクションまでであるため、6自由度の力覚インタラクションへ拡張させるための検討を行った。

第5章「結論」では、本論文をまとめ、本研究の目的である高品質な力覚インタラクションの実現により、今後期待される研究の展開、展望について述べた。

以上のように、本論文では高品質な力覚インタラクションを実現するためのマル

チレート力覚提示システムを提案をした。本研究により，実世界とバーチャル世界の境界を薄め，現実と見紛うほどの力覚インタラクションの実現という目標に貢献し，バーチャルリアリティ，ヒューマンインタフェースの分野での発展が期待できる。

目次

第1章	序論	1
1.1	はじめに	1
1.2	力覚提示システムの概要	3
1.2.1	物理シミュレーション	3
1.2.2	力覚インタフェース	7
1.2.3	力覚レンダリング	10
1.2.4	力覚提示システム	11
1.3	本研究の目的と提案	13
1.4	本論文の構成	13
第2章	マルチレート力覚提示システムと課題	15
2.1	マルチレート力覚提示システム	15
2.2	マルチレート力覚レンダリング	17
2.3	第2章のまとめ	18
第3章	6自由度力覚レンダリングのための中間表現	20
3.1	研究の目的と提案手法の概要	20
3.1.1	中間表現の拡張についての提案	20
3.1.2	システム構成	23
3.2	提案手法の実装	23
3.2.1	6自由度力覚レンダリングのための中間表現	23
3.2.2	6自由度力覚レンダリング	27
3.2.3	摩擦力のレンダリング	29
3.2.4	提案手法の計算手順	32
3.3	評価	33
3.3.1	実験環境	33

3.3.2	6自由度の力覚提示の評価	33
3.3.3	摩擦力提示の評価	36
3.3.4	シミュレーションによる計算時間の評価	37
3.3.5	力覚ポイントの接触点数毎の計算時間の評価	40
3.4	考察	41
3.4.1	プロキシ位置姿勢計算のための回転の重み行列	41
3.5	第3章のまとめ	42
第4章	局所シミュレーションを用いた力覚提示システム	43
4.1	提案手法の概要	43
4.1.1	はじめに	43
4.1.2	関連研究	46
4.1.3	システム構成と処理の流れ	47
4.1.4	記号の表記	48
4.2	物理スレッドの処理	50
4.2.1	近傍物体の探索	50
4.2.2	摂動シミュレーションによる近傍物体のアクセラランスの決定	52
4.3	力覚スレッドの処理	54
4.3.1	3自由度力覚レンダリング	54
4.3.2	局所シミュレーション	56
4.4	大域, 局所シミュレーションの同期処理	57
4.4.1	大域シミュレーションへ局所シミュレーション結果を反映	57
4.4.2	局所シミュレーションへ接触力以外の力の反映	57
4.5	提案システムの評価	58
4.5.1	評価項目	58
4.5.2	実験環境	59
4.5.3	計算機シミュレーションによる力覚レンダリングの評価	59
4.5.4	恒常法による主観評価実験	61
4.5.5	計算時間の評価	65
4.6	考察	69
4.6.1	近傍物体のアクセラランスの効果	69
4.6.2	提案手法の適用範囲	69

4.6.3	6自由度力覚インタラクションへの拡張の検討	70
4.6.4	実装したシステムを利用したインタラクションの例	72
4.7	第4章のまとめ	72
第5章	結論	76
5.1	本論文のまとめ	76
5.2	応用と展望	78
5.2.1	応用	78
5.2.2	展望	79
	参考文献	82

目 次

1.1	コンピュータグラフィックスとのインタラクション	2
1.2	力覚提示システムの構成	3
1.3	ペナルティ法のシミュレータ	4
1.4	接触体積の取得方法 [1] より引用	5
1.5	解析法のシミュレータ	6
1.6	撃力法のシミュレータ	7
1.7	SPIDAR-G6 [2]	9
1.8	Hiro [3] より引用	9
1.9	GyroCube Sensuous [4] より引用	9
1.10	Haptic Ring [5] より引用	9
1.11	力覚レンダリング	10
1.12	ペナルティ法のシミュレータとの接続	12
1.13	解析法のシミュレータとの接続	12
2.1	更新速度の違いによる同期のずれ	16
2.2	力覚レンダリングの分類	17
3.1	提案と従来の中間表現の比較	21
3.2	複数の物体と接触した時の中間表現の例	22
3.3	提案と従来の中間表現による力覚レンダリングの比較	22
3.4	システム構成	23
3.5	6自由度力覚レンダリングのための 中間表現	24
3.6	中間表現の例	26
3.7	力覚ポインタの侵入解除	27
3.8	摩擦力提示計算の手順	31

3.9	実験中の操作	33
3.10	力覚ポインタ y 方向の位置, 加わった力	35
3.11	力覚ポインタ z 軸周りの角度, 加わったトルク	35
3.12	摩擦力提示実験	36
3.13	なぞり操作時の提示力の方向	38
3.14	抗力, 摩擦力, 回転トルクの大きさ	38
3.15	計算時間評価のための シミュレーション構成	39
3.16	力覚提示システムの実時間内計算率	39
3.17	接触点数毎の計算時間	40
3.18	回転の重み行列による侵入解除状態の変化	41
4.1	シミュレーションの分離	44
4.2	近傍物体のアクセラランス	44
4.3	近傍物体の運動と入力する力の関係	46
4.4	局所シミュレーションを適用する範囲の比較	47
4.5	システム構成と処理の流れ	49
4.6	バウンディングボックスでの衝突判定	51
4.7	最近傍点の探索と中間表現	51
4.8	摂動シミュレーションの手順	52
4.9	摂動シミュレーションのための近傍物体に加える摂動力	53
4.10	中間表現の内挿	55
4.11	3自由度力覚レンダリング	55
4.12	構築したバーチャル世界	60
4.13	力覚レンダリングによる力	61
4.14	実験の様子	62
4.15	恒常法による被験者実験結果	64
4.16	近傍物体に接触する物体の影響	65
4.17	立方体数 1 での計算時間	67
4.18	立方体数 1 での計算時間 拡大図	67
4.19	立方体数 10 での計算時間	68

4.20	立方体数 10 での計算時間 拡大図	68
4.21	近傍物体の数の限界	70
4.22	6 自由度力覚インタラクションのための摂動シミュレーションの検討	70
4.23	多数の多面体との力覚インタラクション	73
4.24	多関節剛体とのインタラクション	74
4.25	固有振動の付加による材質感の提示	75

表 目 次

1.1	シミュレータとインタフェースの接続	11
4.1	ハードウェア構成	59
4.2	システムに実装した手法	59
4.3	力覚ポインタが立方体に加えた力積	61
4.4	主観的等価点	63

第1章

序論

1.1 はじめに

近年では計算機能力の向上に伴い、計算機を使用し様々なアプリケーションが生み出されている。これにより、コンピュータグラフィックスとリアルタイム物理シミュレーションを組み合わせて構築されるバーチャル世界も広範囲を人に提示可能となった。そして、人がバーチャル世界とインタラクションすることでバーチャル世界の存在感をより際立たせている。人がバーチャル世界をリアルタイムにインタラクションする方法として、計算機で利用されるマウス、ビデオゲーム用のコントローラ、力覚インタフェースといったヒューマンインタフェースを利用することが挙げられる。特に力覚インタフェースを利用する力覚提示システムはバーチャル世界の操作を直感的に行える(図 1.1)。力覚インタフェースはインタフェース自身が力や位置を変化させることによって、ユーザに力を提示する装置である。現実世界では人が指で物体に触れたとき、指と物体の間には抗力がはたらき、その抗力により物体の形状を力覚によって知覚することができる。同様にバーチャル世界に存在する物体も抗力を指に伝えることによって、物体の形状を知覚できる。さらに、物体の運動をシミュレーションし現実世界と同様の振る舞いをさせることで、物体の慣性や衝突による撃力も知覚することができる。これにより、人はバーチャル世界の物体を現実の物体を扱うかのように、直感的にインタラクションを行うことができる。このように、力覚提示を伴うインタラクションは力覚インタラクションと呼ばれ、設計 [6]、教育 [7]、訓練 [8]、エンタテインメント [9] といった様々な分野で応用されている。

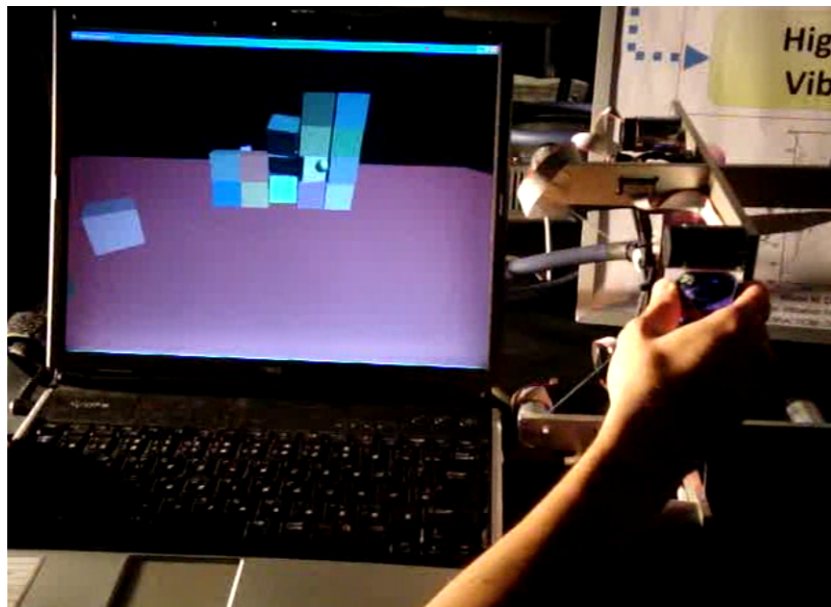


図 1.1: コンピュータグラフィックスとのインタラクション

1.2 力覚提示システムの概要

力覚提示システムは計算機，視覚ディスプレイ，力覚インタフェースの3要素で構成される(図1.2)．計算機はバーチャル世界の物理シミュレーション，力覚提示のための計算である力覚レンダリング，グラフィクスレンダリングを行う．視覚ディスプレイはバーチャル世界の状態をユーザの視覚に提示する．そして，力覚インタフェースはユーザへ力の出力を行う．物理シミュレーションと力覚インタフェースはそれぞれ，様々な手法，機構が提案されている．力覚提示システムはこれらの組み合わせにより，扱えるバーチャル世界の規模，力覚提示の精度が決まる．本論文では剛体の物理シミュレーションと力覚インタフェースを組み合わせたシステムについて議論していく．これ以降は「剛体の物理シミュレーション」は略し「物理シミュレーション」と表記する．

本節では物理シミュレーション，力覚インタフェース，力覚レンダリングについて，それぞれ概要を説明し，これらの要素をまとめた力覚提示システムの構築方法について説明する．

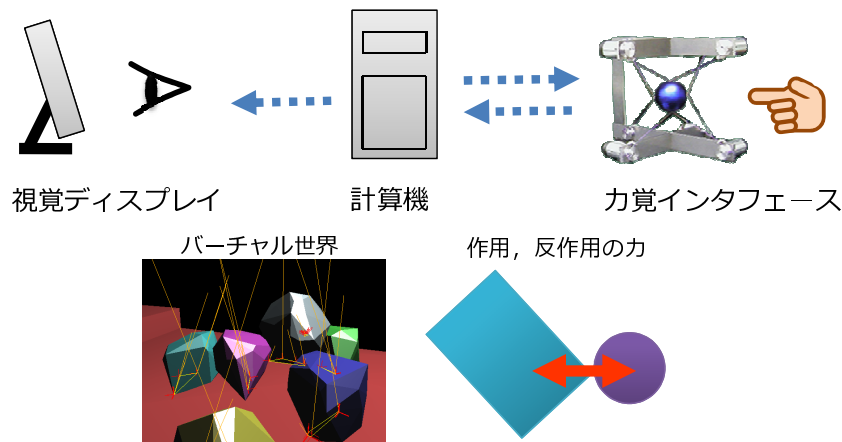


図1.2:力覚提示システムの構成

1.2.1 物理シミュレーション

物理シミュレーションは剛体同士が接触または衝突した際に発生する作用・反作用の力(拘束力)の計算方法で分類できる．また，拘束力の計算は剛体同士の接触

状態の扱い方によって決まるため、衝突判定の考え方も重要となる。以下に物理シミュレーションの拘束力の計算方法と衝突判定の方法について挙げる。

ペナルティ法 [10][11][1] 剛体が他の剛体と衝突や接触をした際、剛体同士をバネダンパモデルでつなぎ、剛体が侵入した量に比例した力を拘束力とする (図 1.3)。拘束力の計算が容易であり、接触点の数を n としたときシミュレーション 1 ステップ当たりの計算量が $o(n)$ と少ない。ペナルティ法ではシミュレーションの更新速度とバネダンパ値により、安定性が決まる。一般に更新速度が速いほど、侵入量を小さくすることができ、バネダンパ値を大きくすることができる。これにより、収束の早い安定なシミュレーションが行える。一方、更新速度が遅いと、侵入量が大きくなるため、発振を押さえるためにバネダンパ値を小さくする必要がある。そのため、収束が遅く不安定になる。つまり、安定なシミュレーションを実現するためには、高速更新が必要となる。しかし、高速更新をさせると時間あたりの計算量が増加するため、大規模なバーチャル世界を構築することは難しい。

上述のようにペナルティ法は剛体同士の侵入量に基づいて拘束力を計算するため、侵入量が取得可能な衝突判定の方法が必要となる。例えば、長谷川ら [1] は拘束力を剛体同士の侵入体積に基づいて計算している (図 1.4)。侵入体積の取得のために、まず剛体同士が侵入し合う共有点を GJK アルゴリズム [12][13] により求める。そして、得られた共有点に基づき Muller と Preparata の双対変換利用する方法 [14] を利用すると侵入形状の頂点が得られるので、侵入形状の体積を求めることができる。

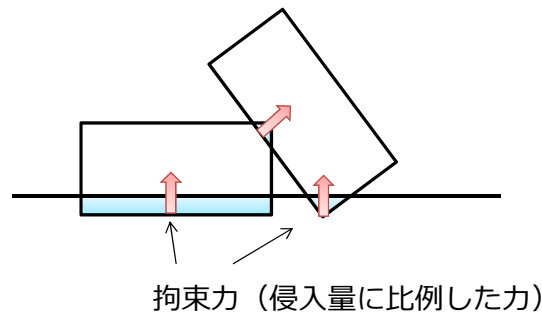


図 1.3: ペナルティ法のシミュレータ

解析法 [15] [16] [17] 剛体が他の剛体と侵入しないように拘束力を計算する方法である (図 1.5)。解析法は剛体の運動方程式を接触している剛体の接触点を基準とし

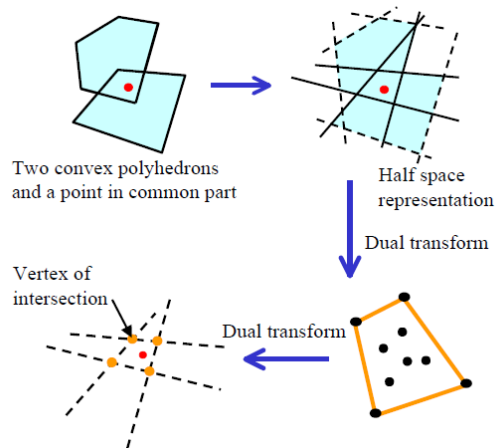


図 1.4:接触体積の取得方法 [1] より引用

た拘束座標系に変換し，速度を拘束した線形相補性問題に帰着させ，求めるべき拘束力を計算する．接触点の数が求める拘束力の数となり，接触点の数を n とすると n 本の 1 次方程式が必要である．一般に，1 次方程式の数は膨大になり，厳密解を解くには $o(n^3)$ の計算量を必要とし，実時間で解くことが困難な場合が多い．そのため，実時間で計算を完了させるために，ガウスザイデル法といった反復法が利用されている．反復法では，解の許容誤差または反復回数で計算量を設定することができるが，解の誤差を小さくするには反復回数を多くする必要がある．そのため，解に精度が求められる，実時間で計算が完了する必要がある等，状況に応じて許容誤差，反復回数の設定が求められる．解析法はシミュレーション 1 ステップ当たりの計算量がペナルティ法よりも多い．そのため，シミュレーションを高速に更新させることが難しいが，ある程度更新速度を落としても安定にシミュレーションを行うことができる．

解析法を利用して拘束力を計算するためには，接触剛体同士の接触点が必要となる．判定のアルゴリズムとして，上述のペナルティ法の項目で紹介した GJK アルゴリズム [12][13] や Lin Canny アルゴリズム [18] が挙げられる．これらのアルゴリズムは判定対象の 2 物体の最近傍点を返すことができる．しかし，これらのアルゴリズムは判定対象の物体が離れている場合のみにしか適応できない．そのため，接触および干渉している物体については，接触していない状態まで物体を移動させてアルゴリズムを適応させる連続衝突判定 (Continuous Collision Detection)[19] が提案されている．上記のアルゴリズムは凸形状のみに適応できる．凹形状で衝突判定を

行いたい場合には凹形状を凸分割し，それぞれに対してアルゴリズムを適応させればよい [20] .

また，凸形状にも対応した衝突判定のアルゴリズムとしてはBVH (Bounding Volume Hierarchies) による判定が提案されている . BVH は物体の形状を階層化近似して判定を行う方法がある . BVH は物体を AABB (Axis Aligned Bounding Box) や OBB (Oriented Bounding Box) , k -DOP (K Discrete Orientation Polytope) , ConvexHull 等のバウンディングボックスで覆い，そのバウンディングボックスの中に複数のバウンディングボックスを再帰的に作成し，物体の近似形状を構築する . 階層構造は木構造になるため，衝突の判定は $o(n \log n)$ で済む . BVH は階層構造が必要のため，オフラインでの前処理が必要であるが，ポリゴン数が多い形状の物体で使用するには非常に有効である . これらの方法は参考文献 [21] にまとめられている .

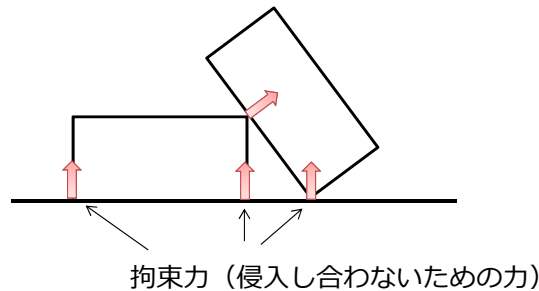


図 1.5: 解析法のシミュレータ

撃力法 [22] 剛体の接触を 2 体の衝突が順番に起こったものとして扱う (図 1.6) . 撃力は接触時間 (time of impact , toi) から求める必要があり , toi によってシミュレーションの刻みが変わる . 複数の剛体が積み重なっている状態では toi が 0 に近づき , シミュレーションが進まなくなる . そのため , リアルタイムシミュレーションに適していない . 計算量は接触点の数 n に対して , $o(n)$ で済むが , 先程述べたようにシミュレーションの刻みは変動する .

撃力法を利用した拘束力の計算は接触点に基づいて行われるため , 解析法の項で説明した衝突判定の方法を利用することができる .

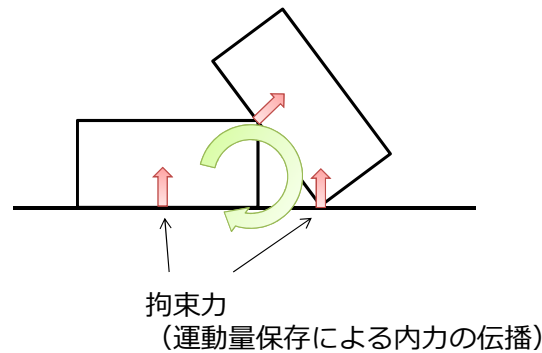


図 1.6: 撃力法のシミュレータ

1.2.2 力覚インタフェース

人が計算機に情報を入力するためのインタフェースの一つとしてマウスが挙げられる。一般にマウスは2次元の平面上の移動距離を情報として計算機に入力するものである。それに対し、力覚インタフェースは計算機に力覚提示部の位置を入力し、力をユーザに出力させることができる。つまり、情報として力を利用することができる。

力覚インタフェースは様々な種類があり、制御方式、提示方式で分類できる [23]。制御方式はインピーダンス制御とアドミタンス制御の2種類がある。インピーダンス制御は力覚提示部の位置を計測し、力を出力する。アドミタンス制御は力覚提示部の力を計測し、位置を出力する。提示方式としては

- ・ 設置

- 把持型 SPIDAR-G6[2](図 1.7), PHANTOM[24], Novint Falcon[9]
- 遭遇型 横小路らのインタフェース [25]
- 装着型 SPIDAR-8 [26], Hiro[3](図 1.8)

- ・ 非設置

- 把持型 Pen de Touch[27], GyroCube Sensuos[4](図 1.9)
- 装着 Gravity Grabber[28], Haptic Ring[5](図 1.10), CyberGrasp[29]

等が提案されている。設置型のインタフェースはインタフェースを台や床等の上に固定して利用する方式で、大きな反力をユーザに提示することが可能である。しか

し、反力が提示できる範囲は利用するインタフェースの可動域の範囲内であるため、作業可能な空間は限られる。一方、非設置型のインタフェースはユーザ自身がインタフェースを把持または装着することが可能であるため、作業可能な空間は設置型のインタフェースと比較して大きい。しかし、提示可能な力の自由度が少ない、大きな反力が提示できないなどの課題がある。以上のように力覚インタフェースには様々なタイプが存在し、互いに長所と短所を持つ。そのため、利用したい環境に応じて適宜選択する必要がある。本研究では高品質な力覚インタラクションが可能なシステムの実現を考えていることから、大きな反力が提示可能、また並進、トルクが出力可能なインタフェースである設置把持型の力覚インタフェースを利用することとし、議論を進めていく。

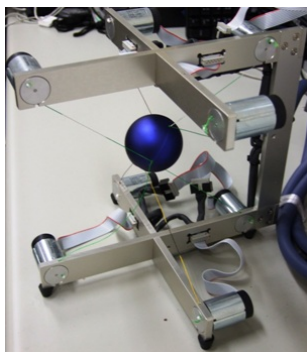


図 1.7:SPIDAR-G6 [2]

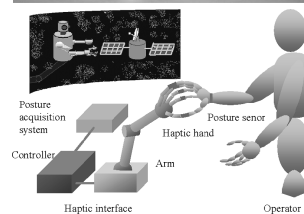


図 1.8:Hiro [3] より引用

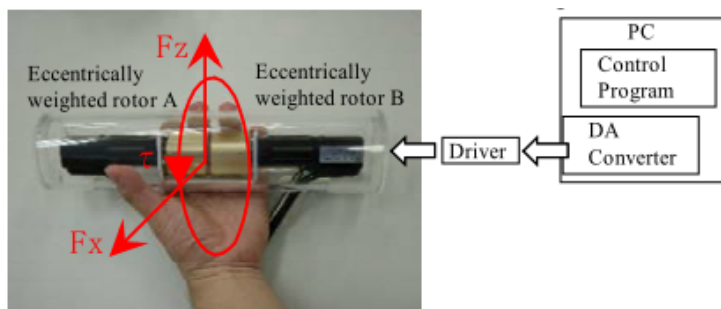


図 1.9:GyroCube Sensuous [4] より引用

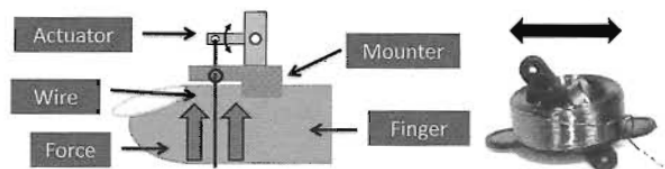
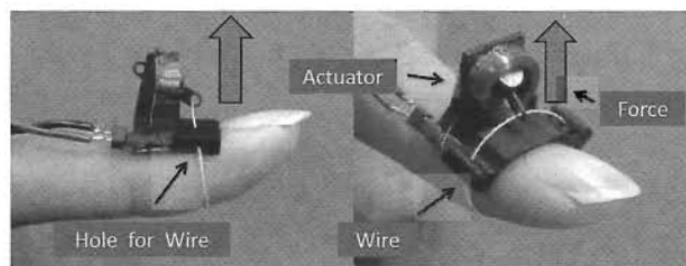


図 1.10:Haptic Ring [5] より引用

1.2.3 力覚レンダリング

力覚レンダリングはユーザへ物体の形状や運動特性を提示するために力覚ポイントと物体に働く力を計算をすることである。現実世界では指で物体に触れた時、指は物体に侵入することはない。それは、指と物体の間に抗力が働くためである。同様にバーチャル世界の物体に力覚ポイントで触れた時に発生する抗力をユーザに提示すれば、ユーザは物体に触れていることを知覚することができる。

提示力は力覚ポイントと物体の侵入の量に比例したものを力とするバーチャルカップリング法 [30] が利用される (図 1.11)。侵入量はインタフェースの位置と物体の位置から求められる。そのため、インタフェースと物体の位置の取得の更新周波数が高いほど侵入量が小さくなるので、安定な力覚提示ができる。一方、周波数が低いと侵入量が大きくなるので、発振を起こし、提示力が不安定となる。不安定な提示力はユーザによるインタフェースの操作性を低下させる。これを防ぐために比例係数を小さくし、提示力を小さくすると実際は硬い物体を提示しているはずのものをユーザは柔らかく感じるという問題が起きる。これまでの研究成果から、人が安定に力覚を感じるための更新周波数の目安として 1kHz 以上が要求されている [30][31]。

バーチャルカップリング(VC)

$$f = -kx - d\dot{x}$$

x:侵入量
k:バネ係数, d:ダンパ係数

- インタフェース位置,
目標位置を高速に取得
x 小 で k 大 → 硬い提示
- インタフェース位置,
目標位置を低速に取得
x 大 で k 小 → 柔らかい提示

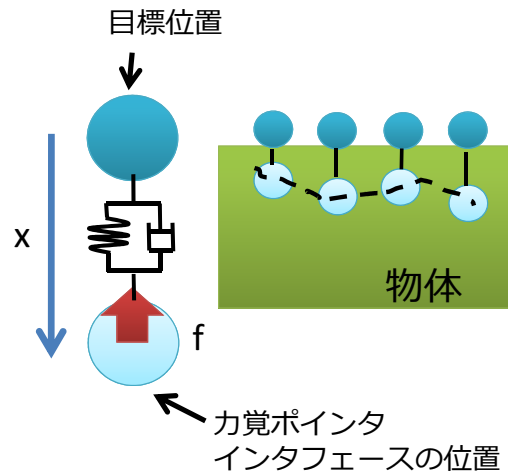


図 1.11:力覚レンダリング

1.2.4 力覚提示システム

本項では、前述した物理シミュレーションと力覚インタフェースを組み合わせ方の方法について述べ、それぞれの特徴について説明する。

ペナルティ法のシミュレータとの接続 [1] ペナルティ法の物理シミュレータは高速に更新することで安定性を保つので、物理シミュレータと力覚インタフェースの間を補間なしに、直接接続(図 1.12) することができ、高精度な力覚提示を行える。しかし、ペナルティ法の物理シミュレータは物体数が増加すると、シミュレーションの計算量が増えるため多くの物体を扱うことができない。多数の物体を物理シミュレーションするためには、解析法の物理シミュレータを利用するほうが望ましい。

解析法のシミュレータとの接続 [1] 解析法の物理シミュレータで安定にシミュレーションをするためには、低速更新が必要であるため、解析法の物理シミュレータと力覚インタフェースを直接つなげると、前述の 1.2.3 項で上げたように安定な力覚を得られない。そこで、安定に力覚提示を行うために、スレッドを物理シミュレーションと力覚レンダリングに分割し、物理シミュレーションを低速に力覚レンダリングを高速に更新するマルチレート力覚提示システムが提案されている(図 1.13)。スレッドを分割することで解析法の物理シミュレータと力覚インタフェースを接続できるようになったが、スレッド分割によりユーザの入力からシステムの応答までに時間遅れがある、提示力に不要な慣性力がでるなど問題がある。マルチレート力覚提示システムは次章で詳しく紹介する。

以上をまとめると、各シミュレータとの接続は表 1.1 の通りとなる。

表 1.1:シミュレータとインタフェースの接続

	ペナルティ法	解析法
更新速度	高速	低速
単位時間当たりの計算量	多い	少ない
インタフェースとの接続方法	直接接続	補間接続
システムの応答性	小さい	大きい

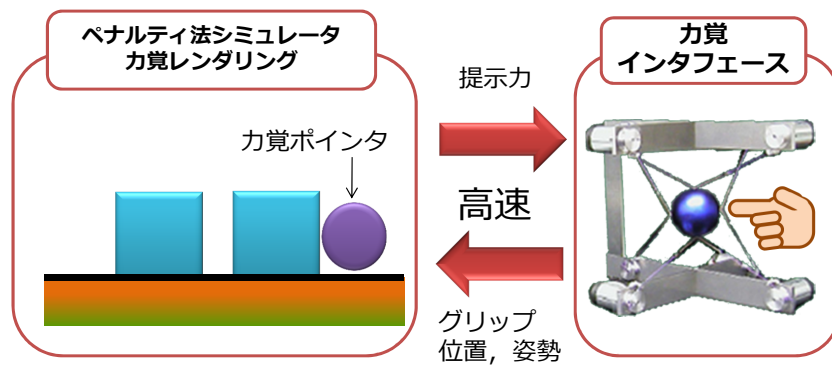


図 1.12:ペナルティ法のシミュレータとの接続

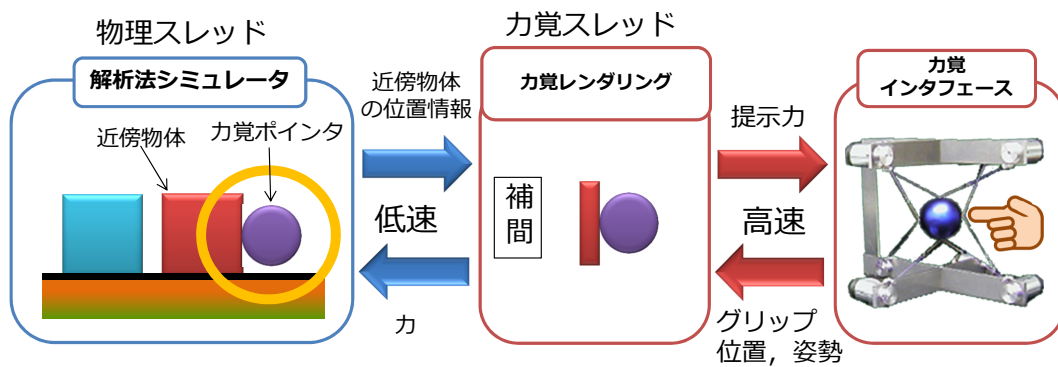


図 1.13:解析法のシミュレータとの接続

1.3 本研究の目的と提案

本研究の目的は大規模で動的なバーチャル世界に対して高品質な力覚インタラクションが可能なマルチレート力覚提示システムを実現することである。本研究では上記の実現のために、「6自由度力覚レンダリングのための中間表現」と「局所シミュレーションを用いた力覚提示システム」を提案する。

「6自由度力覚レンダリングのための中間表現」は6自由度の力覚レンダリングに必要な情報を明らかにすることで、大規模なバーチャル世界に対して不要な慣性力が入らない6自由度の力覚インタラクションを実現する。6自由度の力覚レンダリングに必要な情報は、力覚ポインタ（ユーザがバーチャル世界内で操作する物体）がバーチャル物体と接触した時の、力覚ポインタの接触点群とバーチャル物体に載る平面（力覚ポインタとバーチャル物体間の抗力方向を法線とする平面）である。これらの情報はポリゴンで表される力覚ポインタとバーチャル物体の形状を幾何学的に表したもので、中間表現 [32] と呼ぶ。

「局所シミュレーションを用いた力覚提示システム」は動力学の力覚提示に必要な情報を明らかにし、ユーザの入力に対してシステムの応答に遅れのない正確な力覚提示が可能なシステムを実現する。動力学の力覚提示に必要な情報とは力と加速度の比例係数であるアクセラランスである。

これらの提案ではそれぞれ高品質な力覚インタラクションを実現するために、必要な情報を明らかにしている。「6自由度力覚レンダリングのための中間表現」は力覚ポインタとインタラクションの対象となる物体の形状特徴を表し、その幾何学的な情報を中間表現と呼んでいるが、「局所シミュレーションを用いた力覚提示システム」についても力覚ポインタが加える力と力が加わる物体の動力学的な関係を表しているため、動力学的な中間表現を明らかにしていると言える。そこで、本論文では6自由度力覚レンダリングに必要な情報を幾何学的な中間表現、動力学の提示に必要な情報を動力学的な中間表現と呼ぶ。

1.4 本論文の構成

本論文は5章から構成される。第1章「序論」ではバーチャル世界との力覚インタラクションについて述べ、力覚インタラクションが設計、教育、訓練、エンタテインメント等の分野において応用されていることを述べた。また、力覚提示システム、

特に剛体のバーチャル世界との力覚インタラクションを実現するために必要な要素である物理シミュレーション，力覚インタフェース，力覚レンダリングを挙げ，それぞれについて説明を行った．そして，これらの要素により構成される従来の力覚提示システムについて触れ，特徴，問題点について述べた．最後に高品質な力覚インタラクションのための力覚提示システムの実現が本研究の目的であることを述べた．目的の実現のために，6自由度の力覚提示に必要な情報である幾何学的な中間表現と動力学的な力覚提示に必要な情報たる動力学的な中間表現をそれぞれ明らかにし，それぞれの中間表現を利用した力覚提示システムの提案をすることを述べた．

第2章「マルチレート力覚提示システムと課題」では大規模なバーチャル世界との力覚インタラクションを実現する方法として従来から提案されているマルチレートシステムの研究を紹介する．さらに，従来から提案されている手法について問題点と原因を考察し，高品質な力覚インタラクションを実現するために必要な事項を検討する．

第3章「6自由度力覚レンダリングのための中間表現」では不要な慣性力がでない6自由度の力覚提示を実現するために，6自由度の力覚レンダリングに必要な情報である中間表現を明らかにし，中間表現を利用した6自由度の力覚提示システムを実現する．

第4章「局所シミュレーションを用いた力覚提示システム」では動力学的な力覚提示に必要な情報，アクセラランスを明らかにする．アクセラランスを利用すると，高速に処理を行う力覚スレッドに物理シミュレーションの機能を持たせることができ，ユーザとシステム間の応答性を向上させる力覚提示システムが実現できる．

第5章「結論」では本論文のまとめを行い，応用と展望について述べる．

第2章

マルチレート力覚提示システムと課題

本章では従来のマルチレート力覚提示システムの構成とシステムの構築方法について説明し、その課題について述べる。

2.1 マルチレート力覚提示システム

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。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 [30] 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、安定な硬さの提示の指標として 1kHz 以上の更新 [31] が挙げられている。

- ・ 物理シミュレーションの低速更新 [32][33]

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(図1.13下)。Adachiら[32]、Markら[34]、Hasegawaら[35]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[36]、Akahaneら[37]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[30]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。

これらの手法により、大規模なバーチャル世界を表現可能な力覚提示システムを構築することができた。しかし、これらの手法ではユーザの入力を物理シミュレーションに反映させる時に、プロセスの更新速度の違いからプロセス間の同期に時間を要する。これにより、ユーザの入力が物理シミュレーションに反映されるまで、バーチャル物体の運動は変化しない。例えば静止しているバーチャル物体をユーザが押すような状況(図2.1)では、ユーザが加えた力による物体の動きだしが遅れてユーザにフィードバックされる。そのため、ユーザが物体を押し始める時には物体が動かず、想定よりも大きな力をユーザに提示することになり正確な力覚を提示しているとはいえない。

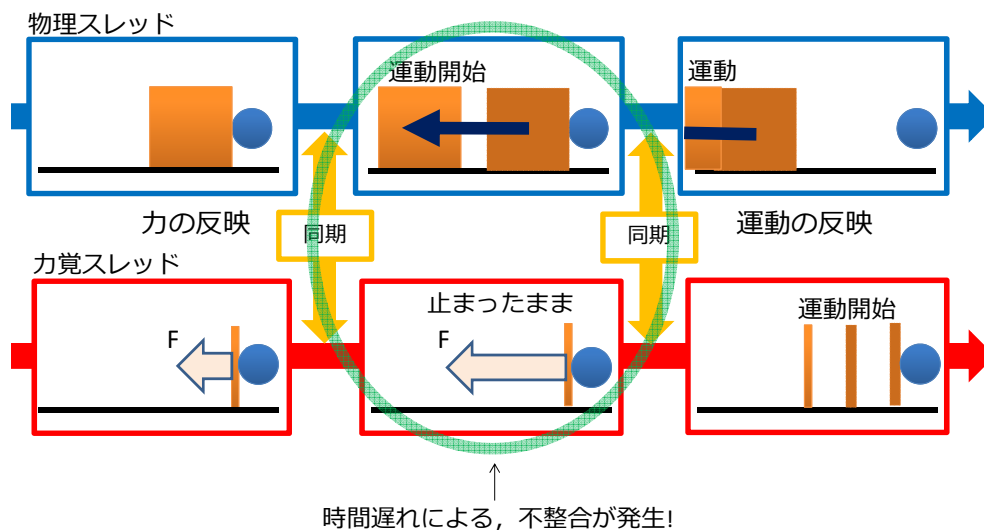


図2.1:更新速度の違いによる同期のずれ

2.2 マルチレート力覚レンダリング

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 [1][38] 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 [39][17] 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．

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 [40]，God Object [41]) 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 2.2.(a))．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 [42] 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，ユーザの操作を障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 [43] 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行う方法を提案している．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

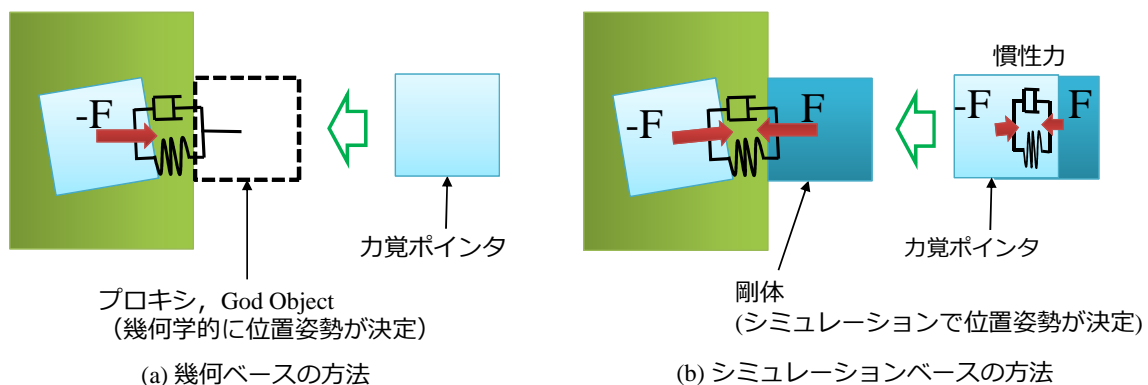


図 2.2: 力覚レンダリングの分類

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 [38] 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 [44][36][37] (図 2.2.(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 [36] 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 [45] 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に有効であるなど手法毎に解決すべき課題がある。

2.3 第2章のまとめ

本節では計算時間を要する大規模なバーチャル世界に対して力覚インタラクションを可能にするマルチレート力覚提示システムについて、従来の研究を挙げ、課題を明らかにした。課題をまとめると、以下の通りとなる。

- ・異なる更新周期のスレッドがそれぞれ同期を行うため、ユーザの入力がシミュレータに反映され、その結果がフィードバックされるまでに時間遅れが起きる。この時間遅れによってユーザとシステム間で時間的な整合性がとれず、正確な力覚提示が行えていない。
- ・力覚提示のための力覚レンダリングには幾何ベースのレンダリングとシミュレーションベースのレンダリングがある。幾何ベースのレンダリングは力覚ポ

インタに質量がないため，バーチャルカップリングによる不要な慣性力の提示はないが，安定な6自由度の力覚提示は実現できていない．シミュレーションベースのレンダリングは6自由度の力覚提示ができているが，質量をもつ剛体と力覚ポイントをバーチャルカップリングしているため，不要な慣性力が提示されてしまう問題がある．

次章以降はこれらの課題に対する解決方法を考え提案し，実装方法と評価について述べる．

第3章

6自由度力覚レンダリングのための中間表現

人間は普通の生活において並進、回転による6自由度の物体の操作を行い、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。

本章では6自由度の力覚を計算する、6自由度力覚レンダリングを実現するための中間表現を明らかにし6自由度力覚提示システムを提案する。そして、提案する中間表現を利用した6自由度の力覚インタラクションシステムを実現し、有効性の評価を行った。

3.1 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する。実現方法として、2.2節で紹介した幾何ベースの6自由度力覚レンダリングに必要な情報である中間表現を明らかにし、6自由度力覚提示システムを提案する。

3.1.1 中間表現の拡張についての提案

2.2節で述べたように、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、本研究では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必要な情報である幾何学的な中間表現を明らか

にし、6自由度の力覚提示を実現する。力覚提示とは力覚ポイントのバーチャル物体への侵入に対する抗力を提示することである。つまり、力覚ポイントがバーチャル物体に侵入しないように力覚レンダリングすれば良い。そして、このような力覚をレンダリングするためには、力覚ポイントが幾何学的にバーチャル物体に侵入していない状態（プロキシ）が求めれば良い。本研究で提案する中間表現とは力覚ポイント（ユーザがバーチャル世界内で操作する物体）がバーチャル物体と接触した時の、力覚ポイントの接触点群とバーチャル物体に載る平面（力覚ポイントとバーチャル物体間の抗力方向を法線とする平面）である（図3.1.(a)、図3.2）。明らかにした中間表現は力覚ポイントと物体間の幾何学的な関係を表し、接触点群が平面に侵入しないための幾何学的な拘束条件を立てることができる。この拘束条件を解くと、プロキシの位置姿勢が求まるので、プロキシと力覚ポイントの状態差から力覚レンダリングを行うことができる（図3.3.(a)）。

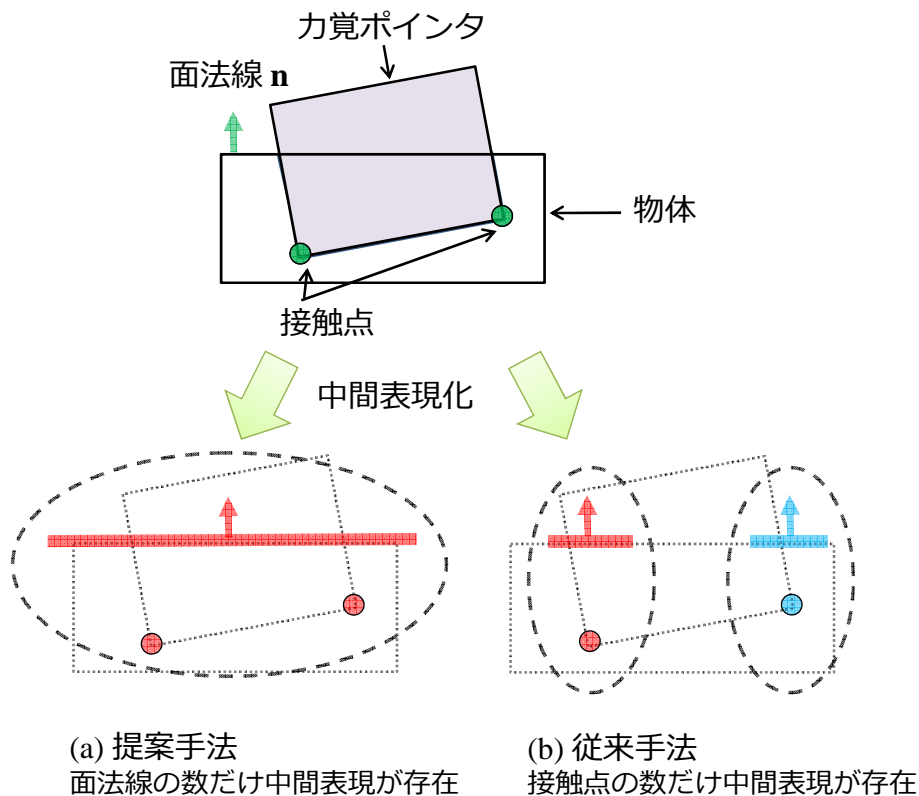


図3.1:提案と従来の中間表現の比較

本研究で明らかにした幾何学的な中間表現は従来の中間表現の形式と似ているが、

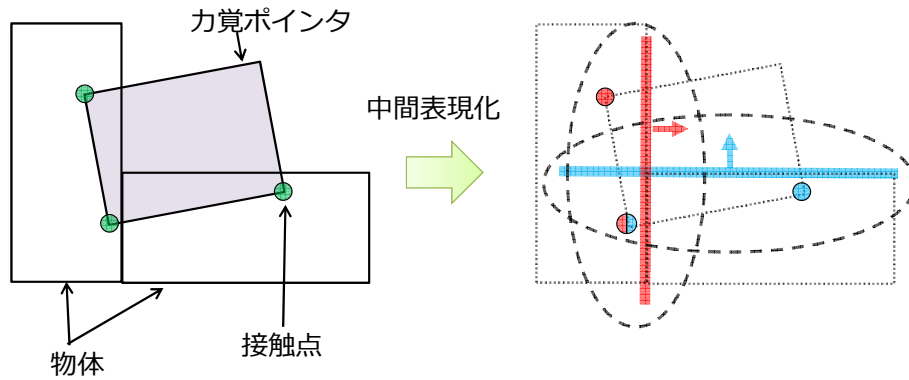
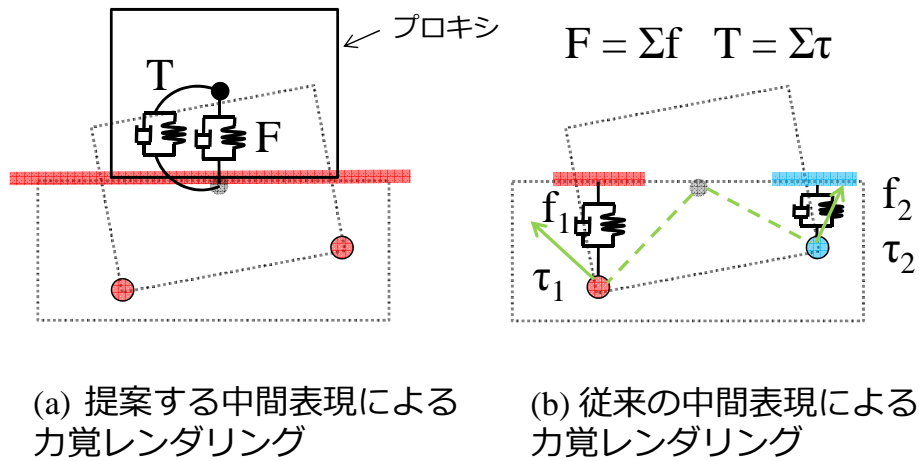


図 3.2: 複数の物体と接触した時の中間表現の例



(a) 提案する中間表現による力覚レンダリング (b) 従来の中間表現による力覚レンダリング

図 3.3: 提案と従来の中間表現による力覚レンダリングの比較

解釈，利用方法が異なる．Berkelman らの方法 [43] は接触点と平面で表される中間表現（図 3.1.(b)）から各接触点の平面への侵入量に基づいて，バネダンパモデルを利用した力覚レンダリングを行なっている．しかし，複数の接触点の侵入量によって計算される力とトルクを合計して，提示すると，提示力が不安定になる（図 3.3.(b)）．本研究で提案する方法は逐次，各接触点に働く力をレンダリングして合算するのではなく，幾何学的な拘束を解いてから，力をレンダリングするため，安定な力覚提示が実現できる．

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 [1] 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．[1] によると 6 個の物体とインタラクションしている状況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．

3.1.2 システム構成

図 3.4 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．

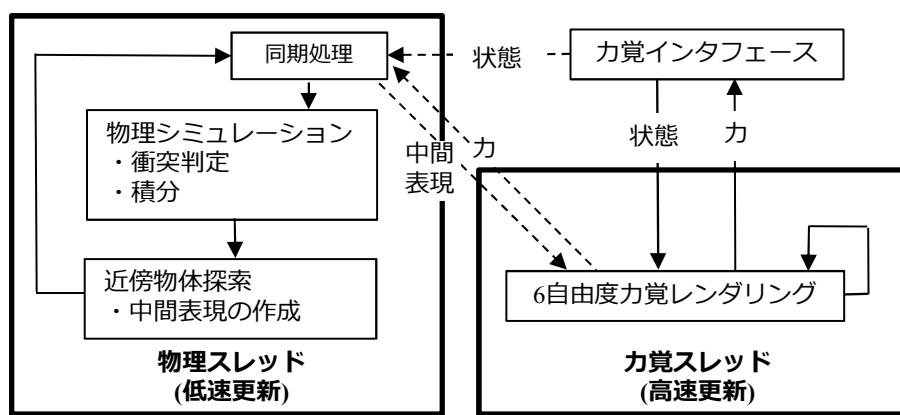


図 3.4: システム構成

3.2 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．

3.2.1 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 [1] で提案されている接触解析の方法を利用する．

中間表現の作成

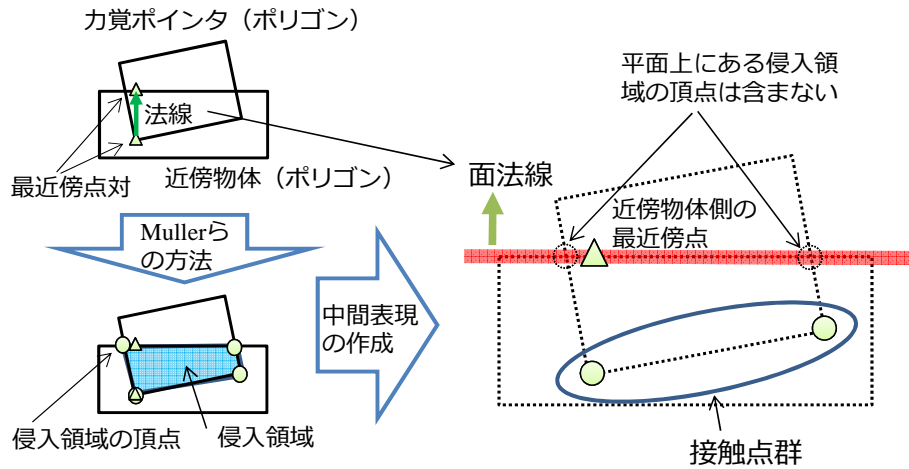


図 3.5:6 自由度力覚レンダリングのための中間表現

力覚ポインタ近傍の物体の探索

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タをユーザが操作し、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（力覚ポインタ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。力覚ポインタ近傍の物体の探索は次のように行う。

まず、力覚ポインタについて、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* [46] で大域接触判定を行う。そして、力覚ポインタ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 [12] や Lin-Canny アルゴリズム [18]、V-Clip アルゴリズム [47] 等を使い、最近傍点対を見つける（図 3.5 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。

侵入領域の頂点の取得

次に力覚ポインタ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 [14] 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对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 3.5 左下)。

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。この平面は前述で得られた力覚ポイントと近傍物体の最近傍点对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。接触点群は前述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 3.5 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 3.6)。

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トの侵入領域の各頂点)

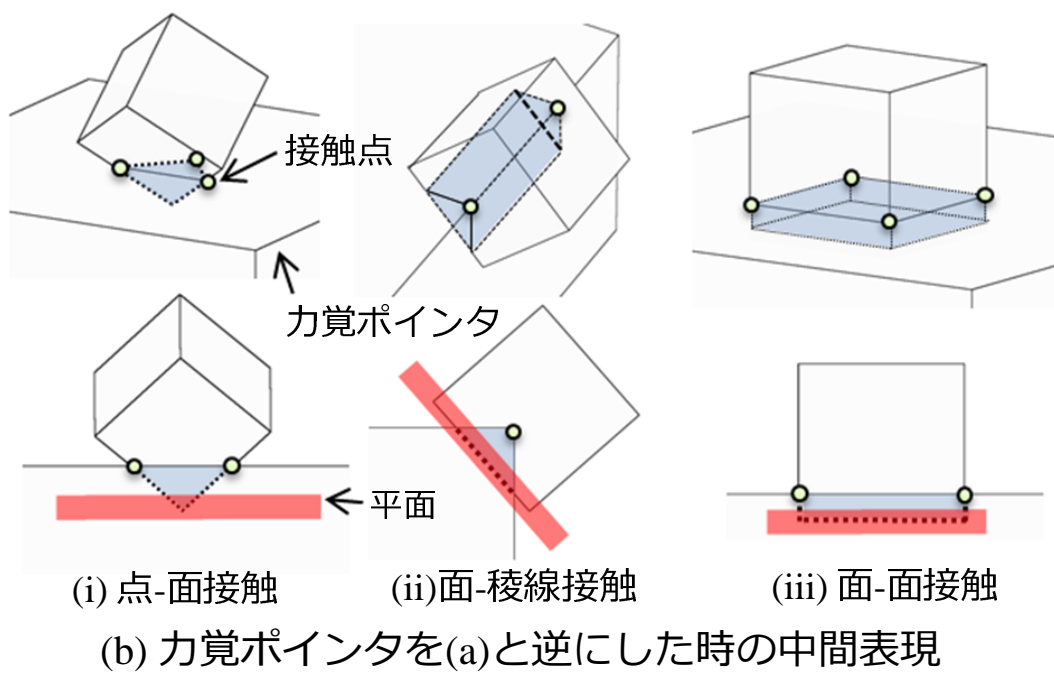
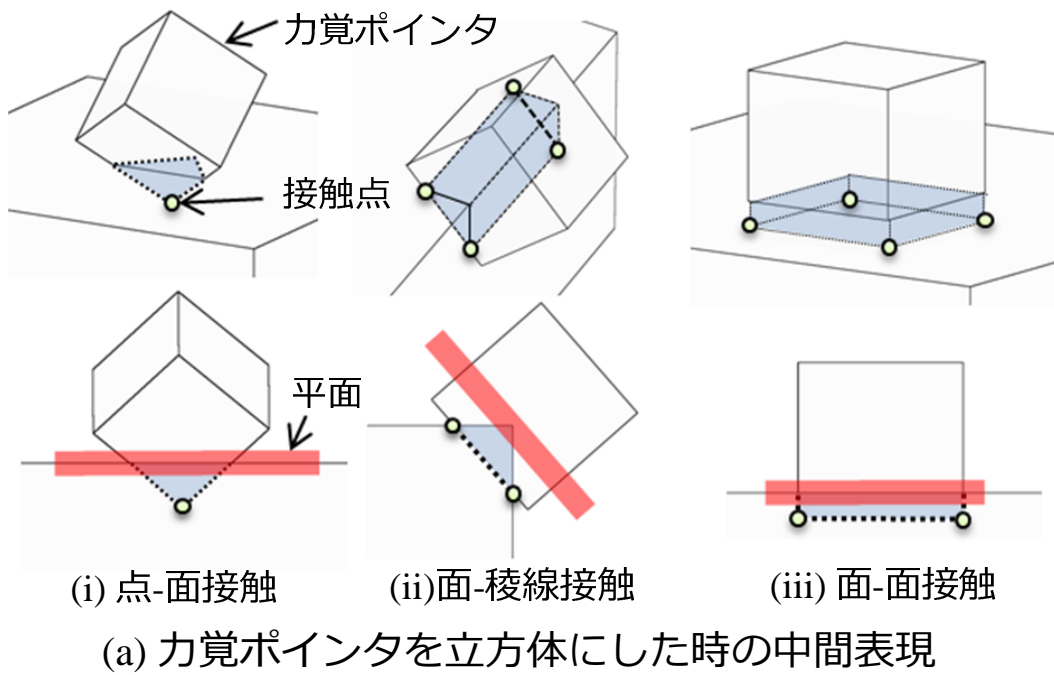


図 3.6: 中間表現の例

3.2.2 6 自由度力覚レンダリング

力覚レンダリングは横山ら [48] 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が操作する物体と力覚ポイント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。

本研究では力覚ポイント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ト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。

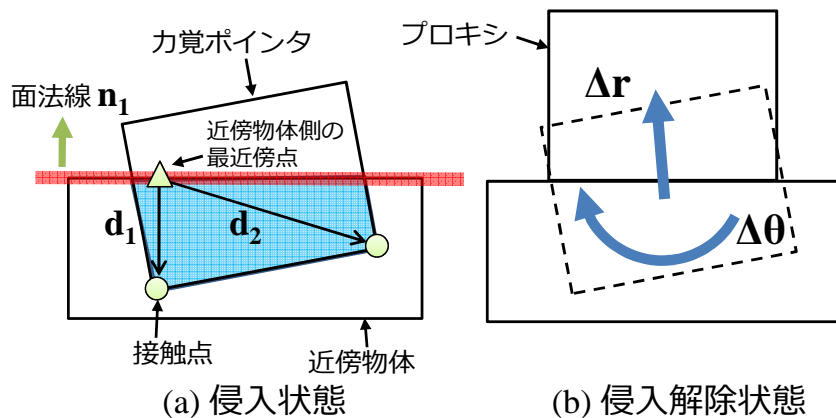


図 3.7: 力覚ポイントの侵入解除

まず、図 3.7.(a) のように力覚ポイントが近傍物体に侵入している状況を考える。平面の法線を $n_i \in \mathbb{R}^3$ とし、3.2.1 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 $d_i \in \mathbb{R}^3$ とする。 i は力覚ポイントが侵入している i 番目の接触点である。ベクトル d_i と平面の法線 n_i との内積をとり、 $d_i \cdot n_i < 0$ 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 $d_i \cdot n_i \geq 0$ となるように力覚ポイントの中心の並進量、回転量を計算する。このとき、力覚ポイント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

$$\Delta\theta \times \mathbf{r}_i + \Delta\mathbf{r} \geq \|d_i \cdot \mathbf{n}_i\| \mathbf{n}_i \quad (3.1)$$

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

$$\Delta\mathbf{r} = \sum_i^n f_i \mathbf{n}_i \quad (3.2)$$

$$\Delta\theta = \sum_i^n f_i \mathbf{R}_w^{-1}(\mathbf{r}_i \times \mathbf{n}_i) \quad (3.3)$$

と表せる．この時，

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

$f_i (> 0)$: 力覚ポインタ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 (3.2)，式 (3.3) を式 (3.1) に代入し，接触点数 n 本分の式をまとめると

$$\mathbf{C}\mathbf{f} \geq \mathbf{d} \quad (3.4)$$

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

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

$$c_{ij} = (\mathbf{r}_i \times \mathbf{n}_i)^T \mathbf{R}_w^{-1}(\mathbf{r}_j \times \mathbf{n}_j) + \mathbf{n}_i^T \cdot \mathbf{n}_j \quad (3.5)$$

である．連立1次不等式である式 (3.4) の \mathbf{f} を求め，式 (3.2)，式 (3.3) 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．

$$f_i^{(k+1)} = \frac{1}{c_{ii}} \left(d_i - \sum_{j=1}^{i-1} c_{ij} f_j^{(k+1)} - \sum_{j=i+1}^n c_{ij} f_j^{(k)} \right) \quad (3.6)$$

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

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. f が収束するように反復回数を設定し, 式 (3.6) に従って 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 f_d , 提示トルク τ_d の計算は侵入解除量 Δr , $\Delta \theta$ を用いて,

$$f_d(t) = k_t \Delta r(t) + d_t \frac{\Delta r(t) - \Delta r(t - \Delta t^L)}{\Delta t^L} \quad (3.7)$$

$$\tau_d(t) = k_r \Delta \theta(t) + d_r \frac{\Delta \theta(t) - \Delta \theta(t - \Delta t^L)}{\Delta t^L} \quad (3.8)$$

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, Δt^L は力覚スレッドの更新時間である.

本項では力覚スレッド1ステップの間の力覚レンダリングについて説明した. 実際は物理スレッド1ステップに対して複数回力覚スレッ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.

3.2.3 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 [40][42] 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 3.8.(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 $p \in \mathbb{R}^6$, プロキシの位置姿勢を $q \in \mathbb{R}^6$ とする. 時刻 $t-1$ の時, 力覚ポイントが物体に侵入し $p_{(t-1)}$ の状態になるとする. すると, 3.2.2 節の方法によりプロキシの位置姿勢 $q_{(t-1)}$ が決まる. この状態から時刻 t に, 力覚ポイントが $p_{(t)}$ へ移動すると中間表現の作成 (3.2.1 節) によって, 新たに接触点 $v_{pi} \in \mathbb{R}^3$ が得られる. そして接触点 v_{pi} を平面に射影したものを v_{psi} , プロキシ位置姿勢 $q_{(t-1)}$ に対して射影したものを v_{qsi} とする.

次に図 3.8.(b) のように力覚ポイントの各接触点に対して摩擦係数 μ 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 3.8.(c)）．摩擦状態の切り替わりは次のように表せる．

$$\mathbf{v}_{qni} = \begin{cases} \mathbf{v}_{qsi} & \text{if } \|\mathbf{v}_{qsi} - \mathbf{v}_{psi}\| \leq \|\mathbf{l}_i\| \\ & \Rightarrow \text{静摩擦} \\ \mathbf{v}_{psi} + \mathbf{l}_i & \text{if } \|\mathbf{v}_{qsi} - \mathbf{v}_{psi}\| > \|\mathbf{l}_i\| \\ & \Rightarrow \text{動摩擦} \end{cases} \quad (3.9)$$

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

$$\Delta\theta \times \mathbf{r}_i + \Delta\mathbf{r} \geq \mathbf{v}_{qni} - \mathbf{v}_{psi} \quad (3.10)$$

となり，式 (3.1) 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 (3.4) の形にまとめることができる．

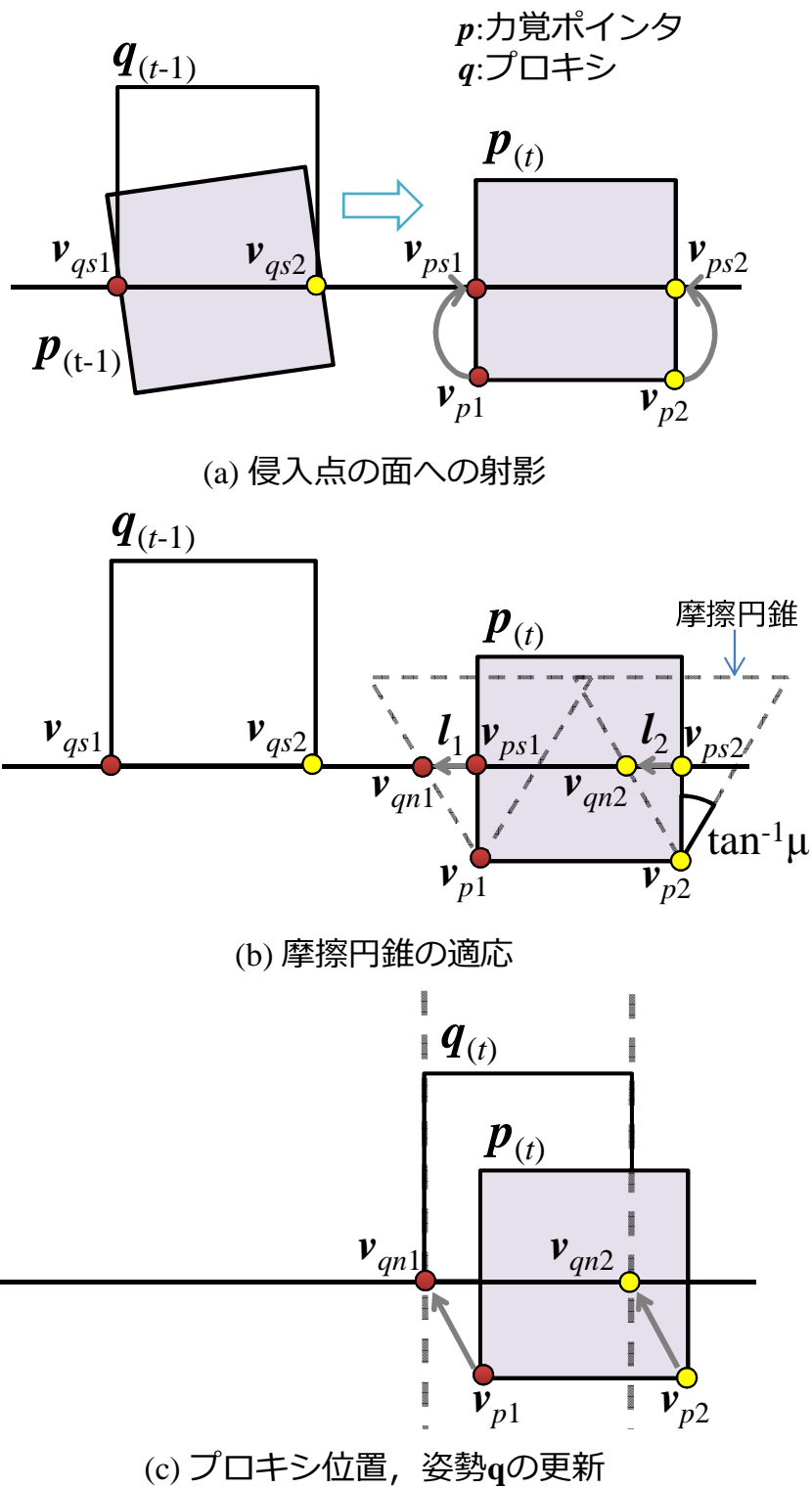


図 3.8: 摩擦力提示計算の手順

3.2.4 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 (3.1) 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 (3.9) に基づいて，各接触点の摩擦状態を決める．
4. 式 (3.10) 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 (3.4) の形にまとめ， f を求める．そして式 (3.2)，式 (3.3) から力覚ポイントの移動量 Δr ， $\Delta \theta$ を求める．
6. 力覚ポイントの移動量 Δr ， $\Delta \theta$ を使い，提示力，トルクを式 (3.7)，(3.8) から計算する．

3.3 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。

3.3.1 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 [2] , 物理シミュレータ Springhead2 [17] 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。

3.3.2 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3.9のように力覚ポイントを操作する。手順は以下の通りである。

- (a) 力覚ポイントを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、力覚ポイントをz軸周り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

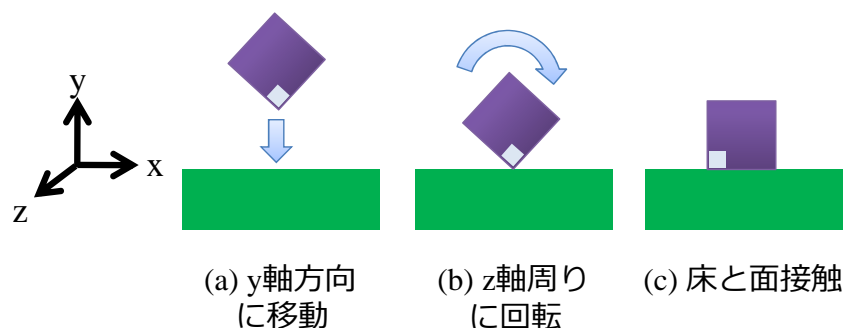


図3.9:実験中の操作

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3.10, 3.11に示す. 力覚ポイントについて, y は y 軸方向の重心位置, θ_z は重心 z 軸周りの回転角度, f_y は y 軸方向に加わる力, τ_z は z 軸周りに加わるトルクを表す. 図3.10, 3.11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 τ_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3.10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)へ向かう状態である. この接触状態の遷移過程では, まず力覚ポイント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)の力覚ポイントが床面に面接触している際にも-0.2Nm程度のトルクが発生している. これは, 力覚ポイント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. 力覚ポイントの姿勢はユーザの操作によって決定されるが, ユーザーは力覚ポイントを精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.

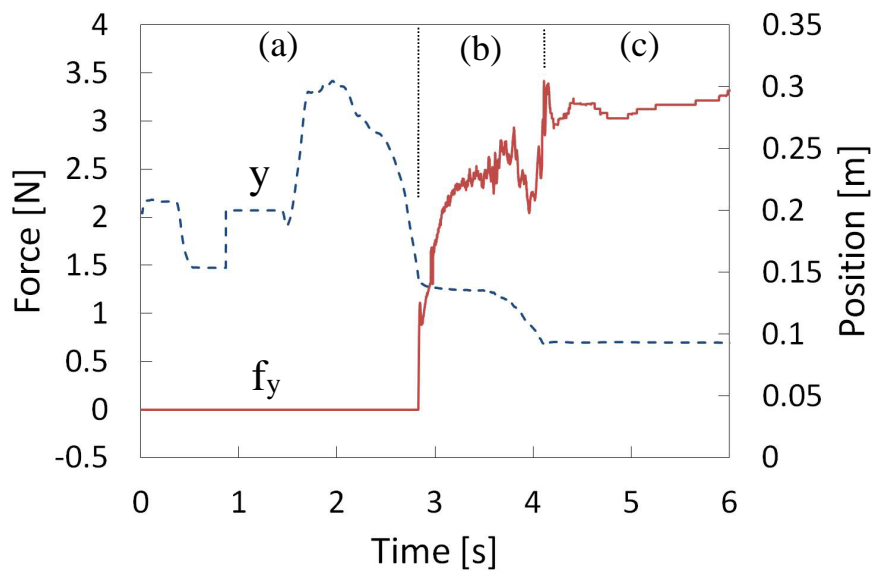


図 3.10: 力覚ポインタ y 方向の位置, 加わった力

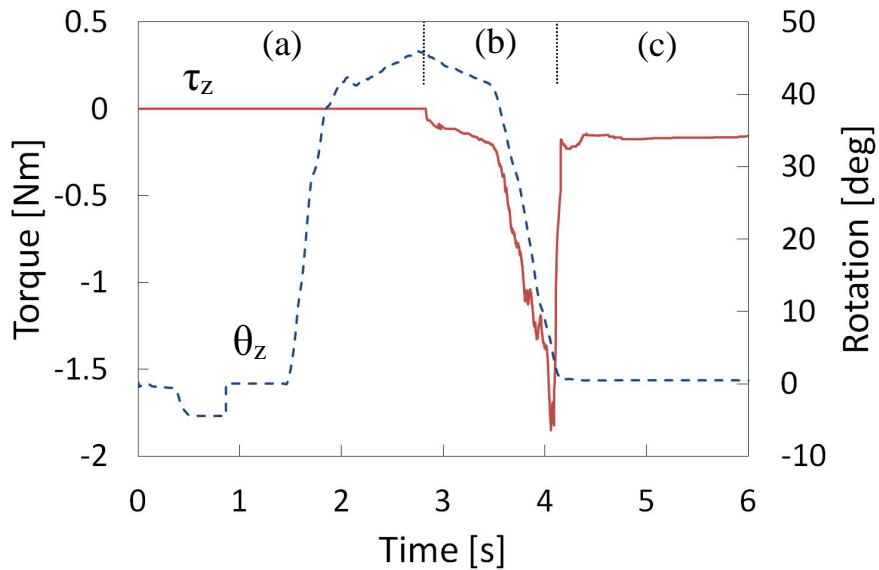


図 3.11: 力覚ポインタ z 軸周りの角度 加わったトルク

3.3.3 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 3.12 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 $k_t = 9000$ N/m， $d_t = 10$ Ns/m，回転は $k_r = 300$ Nm/rad， $d_r = 0$ 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．



図 3.12: 摩擦力提示実験

実験結果を図 3.13，3.14 に示す．図 3.13 は力覚ポインタに加わった力の方向を 100ms のサンプリングでプロットしたもので，図 3.14 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 1ms のサンプリングでプロットしたものである．図 3.13 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 3.14 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 $\mu=0.5$ を除くと，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 3.13 のなぞり操作開始時 (a)，力覚ポイ

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.

3.3.4 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を行った. 評価の方法として, 図 3.15 に示すように直方体の形状を持つ力覚ポインタで, 1列3個の立方体を押し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 [1] (1ms 更新), バーチャルカップリングによる方法 [30] (1ms 更新) である. 時間の計測には Windows の標準 API で μs 単位で計測可能な QueryPerformanceTimer を利用した.

実験結果を図 3.16 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 3.16 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.

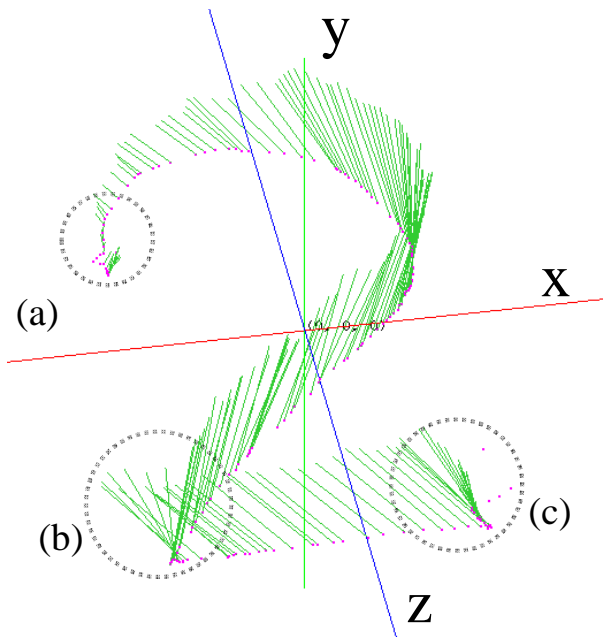


図 3.13: なぞり操作時の提示力の方向

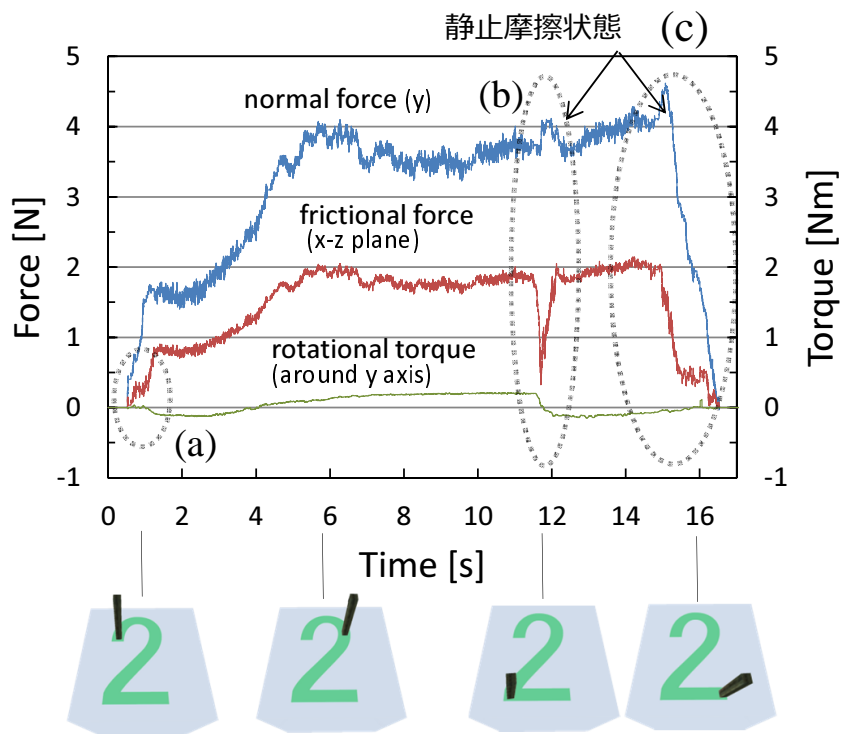


図 3.14: 抗力, 摩擦力, 回転トルクの大きさ

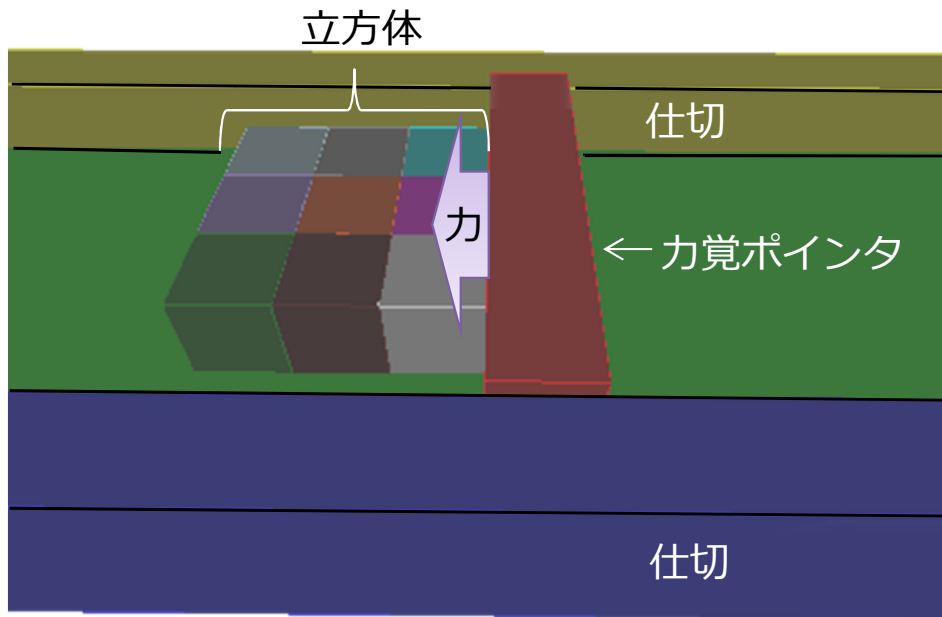


図 3.15: 計算時間評価のためのシミュレーション構成

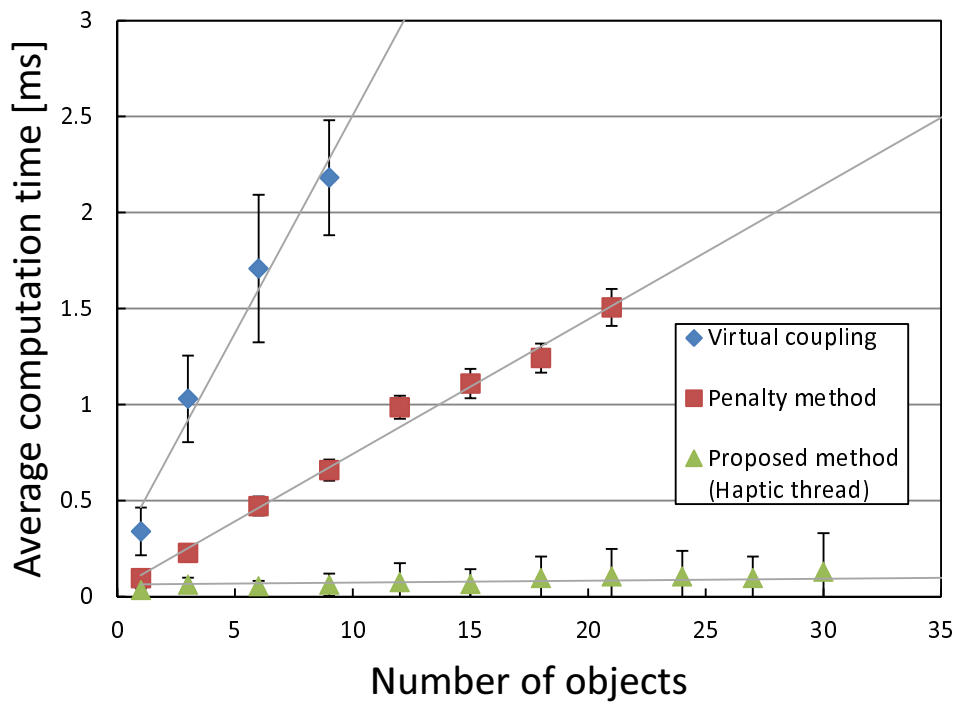


図 3.16: 力覚提示システムの実時間内計算率

3.3.5 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五角柱とし、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 3.17 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 3.17 右に示す。横軸は接触点の数、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。

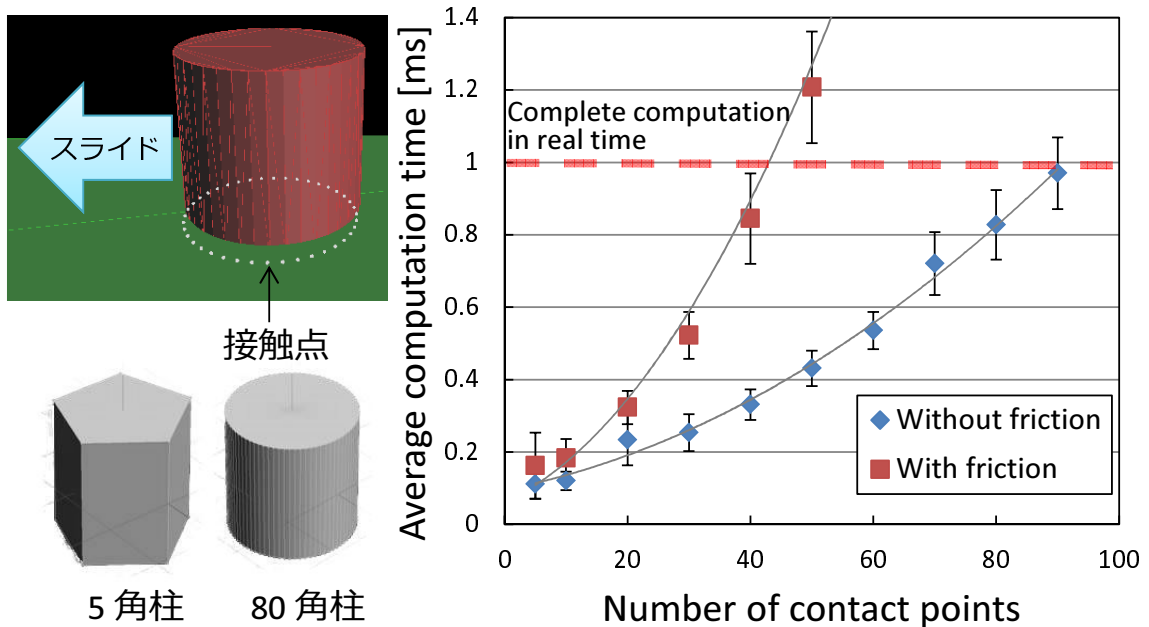


図 3.17:接触点数毎の計算時間

3.4 考察

3.4.1 プロキシ位置姿勢計算のための回転の重み行列

提案手法で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．力覚ポイント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 3.18.(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 3.18.(b) の部分）が起きる．特に後者の場合，力覚ポイント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，力覚ポイントについて侵入の可能性の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トを適当に回転させ，侵入の可能性の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

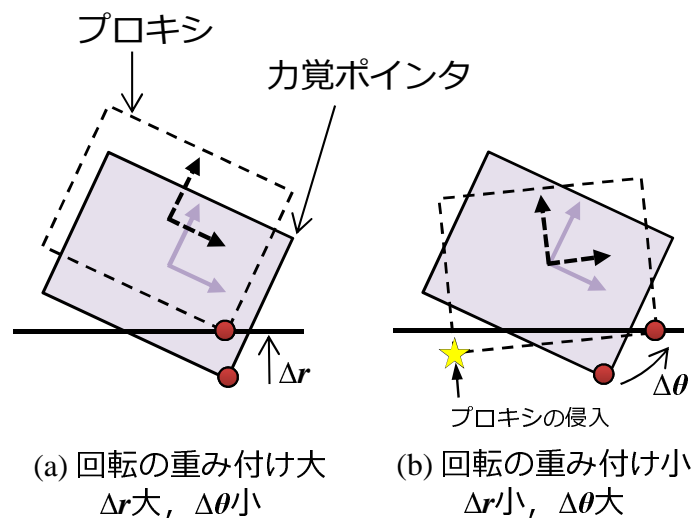


図 3.18: 回転の重み行列による侵入解除状態の変化

ダンパ係数を大きくすることでプロキシが侵入す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し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。

3.5 第3章のまとめ

本章ではマルチレート力覚提示システムにおいて不要な慣性力が提示されない6自由度力覚レンダリングに必要な情報を明らかにした。力覚レンダリングに必要な情報は力覚ポインタが物体に侵入した際に、侵入解除状態（プロキシ）を求めるための条件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え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。

提案手法の実装のために構築した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、2.1節で説明したように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している。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ついては考慮されていない。次章ではユーザとシステム間の応答性について考え、ユーザの入力に対して応答性のあるシステムについて提案する。

第4章

局所シミュレーションを用いた力覚提示システム

本章では動力学の力覚提示に必要な情報，動力的な中間表現を明らかにし，ユーザの力の入力に対して，応答性のあるマルチレート力覚提示システムを実現する．

4.1 提案手法の概要

4.1.1 はじめに

ユーザの入力をスレッドの同期待ちなしに物理シミュレーションに反映させるためには，2.1節で説明したように，物理シミュレーションの更新周期を力覚インタフェースの制御周期に合わせなければならない．しかし，バーチャル世界の規模が大きくなるとシミュレーションの刻み幅内で計算が終わらず，実時間でのインタラクションができない．

そこで，大規模なバーチャル世界上での力覚提示を実現するために，スレッド分割したシステムの力覚スレッドにおいて局所的な物理シミュレーション（以下，局所シミュレーションと呼ぶ）を行う手法を提案する．提案手法では，図4.1のようにバーチャル世界の物理シミュレーションを低速更新の大域シミュレーションと高速更新の局所シミュレーションに分離する．局所シミュレーションの更新を力覚インタフェースの制御周期に合わせることで，ユーザの入力をスレッドの同期待ちなしに物理シミュレーションに反映させることができる．

また，本研究では力覚ポインタ近傍の物体，近傍物体について力覚ポインタが加える力と近傍物体の加速度の比例係数であるアクセラランスを利用し，局所シミュレーションを行う．局所シミュレーションにアクセラランスを利用することで，局

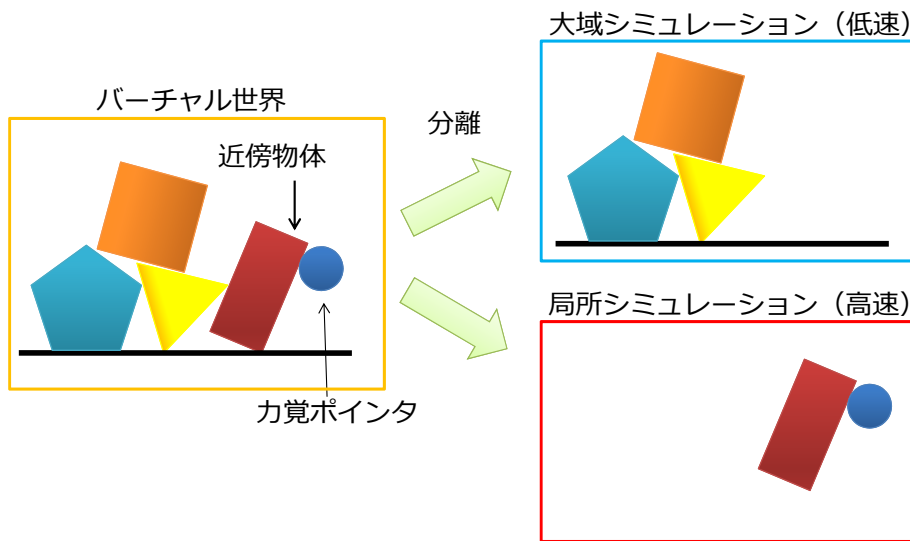


図 4.1: シミュレーションの分離

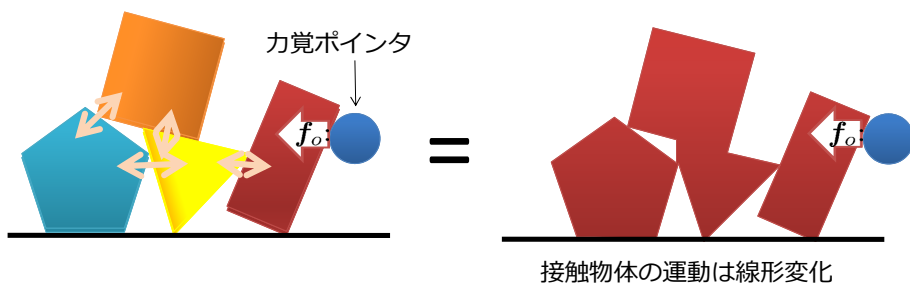


図 4.2: 近傍物体のアクセラランス

所シミュレーションの対象となる物体を力覚ポインタの近傍物体のみに絞ることができる。その理由として、図 4.2 のように力覚ポインタが相互に接触している物体に対して力を加える状況を考える。力覚ポインタが加える力 f_o と近傍物体の運動に線形性を仮定すると、近傍物体の運動はメカニカルインピーダンスを用いて、

$$\begin{aligned}
 M\dot{u} + Bu + K \int u dt + f_e &= \begin{pmatrix} f_o \\ \mathbf{r} \times f_o \end{pmatrix} \\
 &= J_h f_o
 \end{aligned} \tag{4.1}$$

と書ける。ただし、

M, B, K : メカニカルインピーダンス行列 (6×6 行列)

u : 近傍物体の速度, 角速度 (6 次元ベクトル)

f_o : ユーザが加える力 (3 次元ベクトル)

f_e : f_o 以外の外力 (6 次元ベクトル)

r : 近傍物体重心からの f_o の作用点

J_h : f_o を剛体に加える力とトルクに変換する行列 (6×3 行列)

である. 実際, シミュレーションを行う際には式 (4.1) を時間方向に離散化し, 近傍物体の運動を求める. このときの刻み時間は約 0.05s と小さいので, 式 (4.1) を積分したときに 2 次以上の項となる $u, \int u dt$ を定数として扱うことができる. そこで式 (4.1) を変形し, これらの定数をまとめると,

$$\begin{aligned}\dot{u} &= M^{-1} J_h f_o - M^{-1} (Bu + K \int u dt + f_e) \\ &= A f_o + b\end{aligned}\quad (4.2)$$

となる. A は近傍物体のアクセランスで質量慣性成分を表す 6×3 行列, b は重力等の外力 (ユーザが加える力によって変化しない力) から由来する加速度項 (6 次元ベクトル) である. 式 (4.2) では $f_o = (0 \ 0 \ 0)^t$ の場合, b のみが加速度変化に影響する. 一方, $f_o \neq (0 \ 0 \ 0)^t$ の場合は, 加速度項に加え, アクセランスによって加速度が変化する. つまり, アクセランスと加速度項が近傍物体の動力学特性を表しているといえる. アクセランス, 加速度項が求まり, 力覚ポイントが加える力 f_o さえわかれば, 近傍物体がその他の物体の接触から受ける接触力がわからなくても, 近傍物体の運動を正しく求めることができる. つまり, アクセランスと加速度項が動力学的な中間表現である. そのため, 図 4.3 のように, あらかじめ物理スレッド側でアクセランスを求め, 力覚スレッドに送り, 局所シミュレーションに利用することで, 近傍物体の運動のみをシミュレーションすることができる. そこで, アクセランスを求めるための方法を考える. 考えられる方法は解析的に求める方法とシミュレーションにより求める方法の 2 通りがある. 解析的に求める方法は接触グループ全体の質量慣性行列を求め, その逆行列をアクセランスとして導く方法である. しかし, 毎ステップ接触グループの物体を特定し, 質量慣性行列を計算することは非効率である. シミュレーションにより求める方法は接触グループに適当な力を加え, 速度変化を観察し, 質量慣性行列を逆問題的に求める方法である. この方法は, 加える力の方向を適切に設定する必要があるが, シミュレーションによって自動的に求めることができる. そこで本研究ではシミュレーションによ

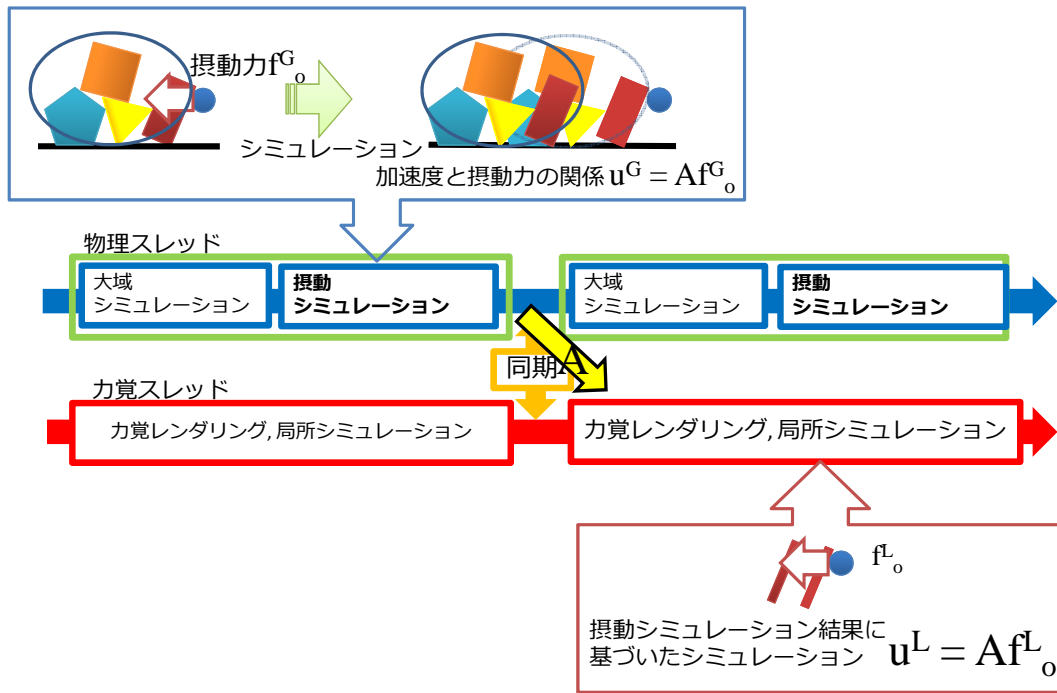


図 4.3: 近傍物体の運動と入力する力の関係

る方法（振動シミュレーション）によって，近傍物体のアクセランスを求め利用する．

この方法は近傍物体が他の物体と接触していたり，リンクやバネ，ダンパで接続されている場合でも，接触状態や接続が変化しない限り線形性の仮定は成り立つ．静止摩擦と動摩擦の切り替えりや接触の増減がある場合は線形性は成り立たないが，そのような変化は大域シミュレーションのステップでのみ起こると考え，局所シミュレーションでは扱わない．本研究では大域シミュレーションと局所シミュレーション間で整合性を保つ方法として接触状態の変化や静止摩擦と動摩擦の切り替えりにより発生する力を大域シミュレーションにて計算を行い，その力を局所シミュレーションに反映させる．詳しくは 4.4.2 項で説明する．

4.1.2 関連研究

関連研究として，Glondou ら [49] が物理シミュレーションを分離し，力覚ポイント近傍の物体を局所的にシミュレーションする方法を提案している．Glondou らは近傍物体を特定の数に限り，局所シミュレーションを行なっているが，接触する物体が

増加すると、計算量が増加し、実時間での力覚インタラクションが難しくなる(図4.4)。一方、提案手法では局所シミュレーションの対象となる物体は力覚ポイント近傍の物体のみとなるので、Glondouらの方法に比べ、計算量の増加を抑えることができる。

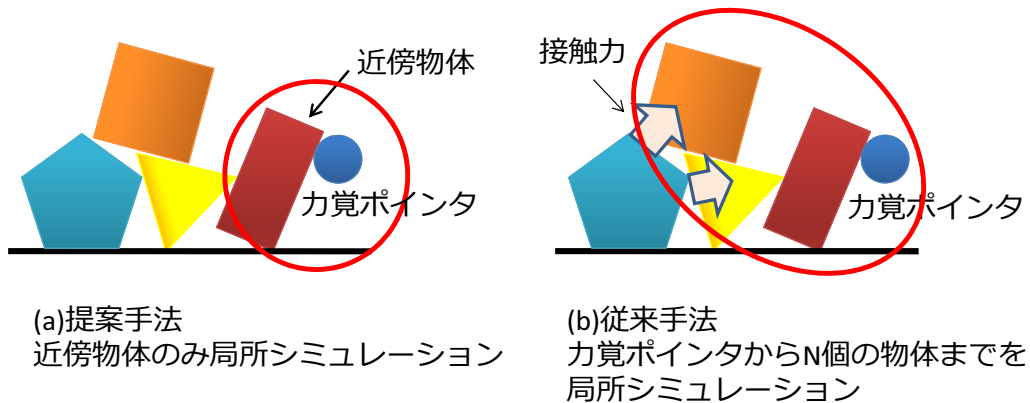


図 4.4:局所シミュレーションを適用する範囲の比較

4.1.3 システム構成と処理の流れ

上述の提案手法を実現するためのシステム構成と処理の流れを図4.5に示す。また、以下に各スレッドの処理について説明する。

・物理スレッド

1. 大域シミュレーション
バーチャル世界全体の物理シミュレーションを行う。
2. 近傍物体の探索(4.2.1項)
力覚ポイントに対して近傍の物体を探索する。
3. 近傍物体のアクセラランスの決定(4.2.2項)
摂動シミュレーションを行い、近傍物体のアクセラランスを計算する。
4. 局所シミュレーションとの同期(4.4.1項)
局所シミュレーションの結果を大域シミュレーションへ反映させる。

・力覚スレッド

- a. 力覚ポインタの状態の更新
力覚インタフェースから力覚ポインタの位置，姿勢を受信する．
- b. 力覚レンダリング（4.3.1項）
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．
- c. 局所シミュレーション（4.3.2項）
力覚ポインタが近傍物体に加えた提示力と，物理スレッドで求めたアクセラランスに基づいて，近傍物体の位置，姿勢，速度，角速度を更新する．
- d. 物理スレッドが終わるまで b～d の処理を繰り返す．
- e. 大域シミュレーションとの同期（4.4.2項）
近傍物体が受ける接触力以外の力を局所シミュレーションに反映させる．

大域，局所シミュレーションの同期は物理スレッドのステップが終わる時点で行う．物理スレッドの更新周期に対する力覚スレッドの更新周期の比率を n とすると，力覚スレッドステップを n 回行っている間に物理スレッドステップを 1 回行い，シミュレーションを同期させる．

4.1.4 記号の表記

本論文では各スレッドステップで使用する記号を以下のように定義する．

G : 物理スレッドで扱う変数の添え字

L : 力覚スレッドで扱う変数の添え字

i : 第 i 回目の物理スレッドステップ

i, j : 第 (i, j) 回目の力覚スレッドステップ

(第 i 回目の物理スレッドステップを行っている間の第 j 回目の力覚スレッドステップ．ただし， $0 \leq j < n$ で物理スレッドと同期後に 0 となる．)

例えば，第 i 回目の物理スレッドステップで大域シミュレーションを行い，ある物体の位置，姿勢 $s_{(i-1)}^G$ を更新したとすると $s_{(i)}^G$ となる．また，第 (i, j) 回目の力覚スレッドステップで局所シミュレーションを行い，近傍物体の位置，姿勢 $s_{(i,j-1)}^L$ を更新したとすると $s_{(i,j)}^L$ となる．

次節からは各スレッドの処理の詳細について述べる．

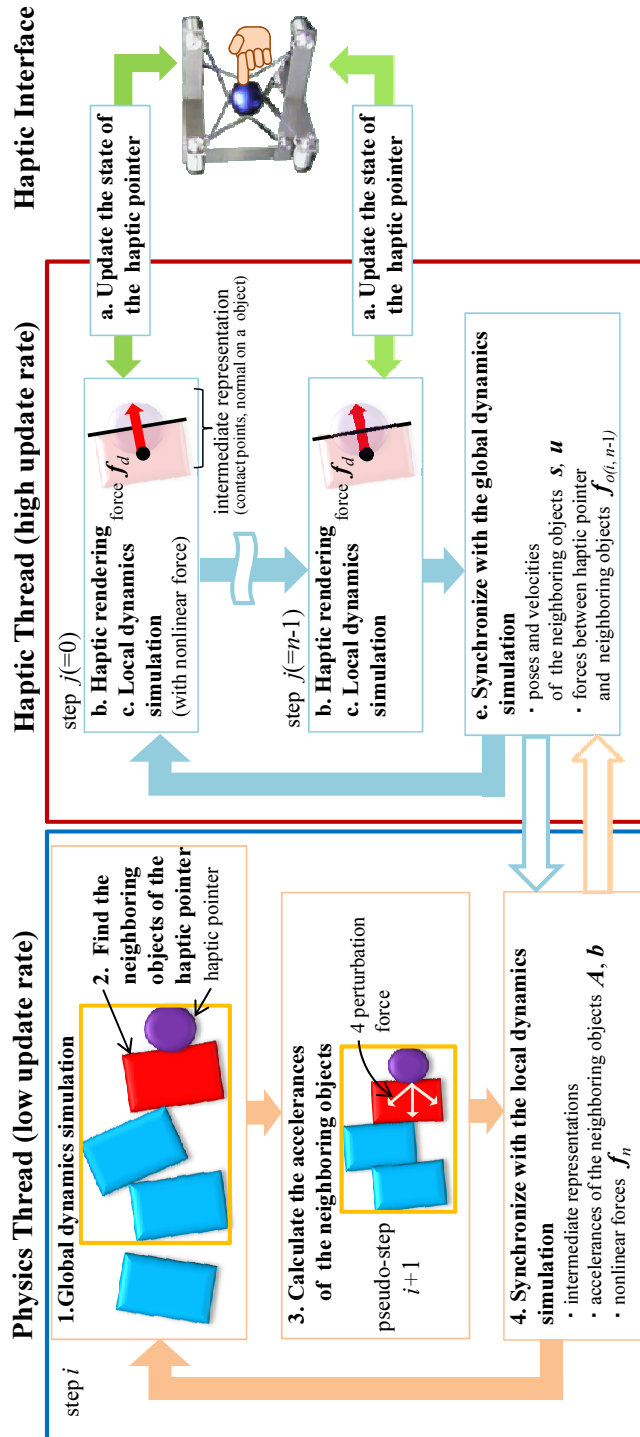


図 4.5: システム構成と処理の流れ

4.2 物理スレッドの処理

4.1.2項で述べたように、近傍物体を正確に局所シミュレーションさせるには近傍物体に接触する物体からの接触力を考慮する必要がある。本研究では近傍物体に加わる接触力を考慮するために、通常の大域シミュレーションからシミュレーションをさらに1ステップ進め(摂動シミュレーション)、近傍物体の速度、角速度変化から近傍物体のアクセラランスを計算する。そして、得られた近傍物体のアクセラランスを局所シミュレーションで使用することで、近傍物体に接触する物体からの接触力を考慮する。本節では近傍物体の探索、局所シミュレーションで使用する近傍物体のアクセラランスの決定方法について説明する。

4.2.1 近傍物体の探索

はじめに、バーチャル世界において力覚ポインタ近傍の物体を決定する方法について説明する。近傍物体の探索には、物理シミュレーションの衝突判定のアルゴリズムを使用する。衝突判定のアルゴリズムは大域判定(Broad Phase)と局所判定(Narrow Phase)とに大別されている[21][47]。大域判定は主に計算コストを必要とする局所判定の前段階として行われ、衝突判定全体としての計算量を削減を目的とし、局所判定では1.2.1項では述べたように剛体同士の接触状態を調べる。

本研究では大域判定では実装が容易であるAABBを用いたSweep and prune[15]を行い、局所判定ではGJKアルゴリズムにより力覚ポインタ近傍の物体を判定することにする。近傍物体判定の流れを示す。

1. バウンディングボックスでの大域判定

バーチャル世界内で力覚ポインタのバウンディングボックスとその他の物体のバウンディングボックスが干渉していれば、その物体を局所シミュレーションの対象とする(図4.6)。局所シミュレーションの対象となった物体は力覚ポインタと直接接触している可能性があるので、2を行う。また、上述のように力覚ポインタのバウンディングボックスは物体を局所シミュレーションの対象とするかの判定を行っているため、ある程度の大きさを持たせることが望ましい。

2. GJKアルゴリズムによる局所判定

力覚ポインタの近傍と判定された物体のみについて，GJK アルゴリズムを利用して力覚ポインタと近傍物体の最近傍点对を探す．得られた最近傍点对の距離と最近傍点对を結ぶ単位ベクトルは力覚レンダリングに使用するので，スレッド間同期のために保存しておく．この時，保存の形式として，Adachiら [32] が提案する中間表現 (intermediate representation) を利用する．中間表現は近傍物体を面で力覚ポインタを点で近似表現させるものである．このようにすることで，スレッド間の通信に負荷を軽減させる．

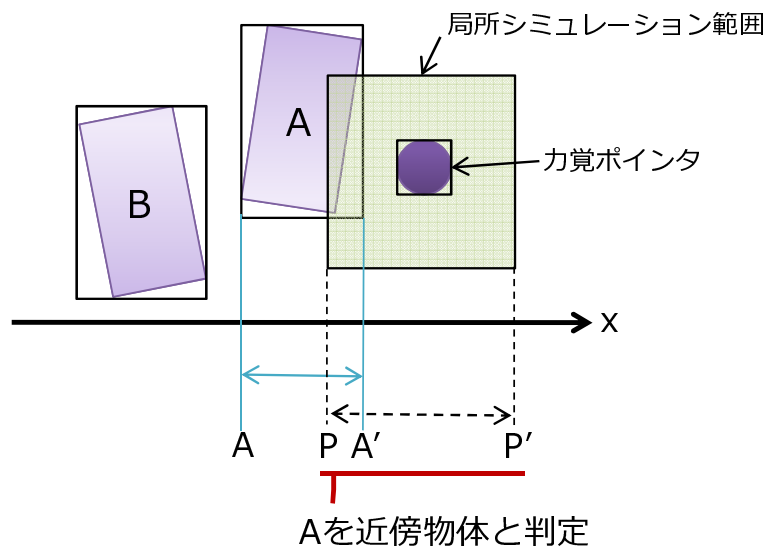


図 4.6: バウンディングボックスでの衝突判定

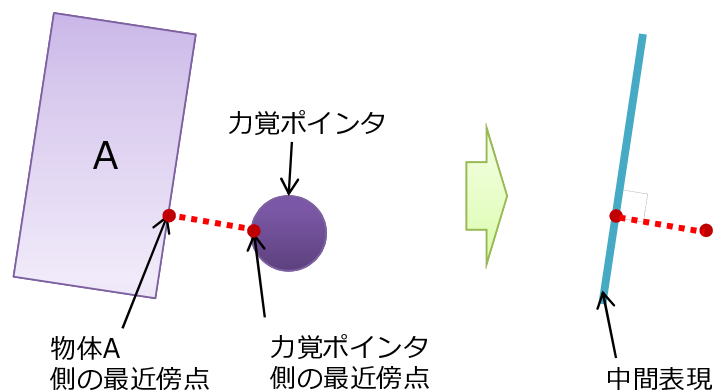


図 4.7: 最近傍点の探索と中間表現

4.2.2 摂動シミュレーションによる近傍物体のアクセラランスの決定

本研究では大域シミュレーション後にさらに1ステップシミュレーションを進める摂動シミュレーションを4回行うことで近傍物体のアクセラランス A と重力等の外力に由来する加速度項 b を決定する (図 4.8) . ここでは第 i 回目の物理スレッドステップでのアクセラランスの決定について説明する .

力覚ポイントが近傍物体に加える力を $f_{o(i+1)}^G = (0, 0, 0)^t$ と設定して摂動シミュレーションを実行すると $u_{0(i+1)}^G$ が得られる . ここで , 大域シミュレーションの刻み時間を Δt^G として , 式 (4.2) を差分形にすると ,

$$u_{(i+1)}^G = u_{(i)}^G + (A_{(i+1)} f_{o(i+1)}^G + b_{(i+1)}) \Delta t^G \quad (4.3)$$

となる . $u_{0(i+1)}^G, u_{(i)}^G, f_{o(i+1)}^G, \Delta t^G$ は既知なので式 (4.3) に代入すると , $b_{(i+1)}$ を求めることができる . 次に第 $(i-1, n-1)$ 回目の力覚スレッドステップから受信したユーザが実際に入力した力 $f_{(i-1, n-1)}^L$ を摂動力 $f_{o1}^G = f_{(i-1, n-1)}^L$ とし , その単位ベクトルを $e_1 (= f_{o1}^G / |f_{o1}^G|)$ とする . そして , e_1 に相互に直交する任意の単位ベクトル2本 e_2, e_3 を求める . 直交条件は以下の通りである .

$$\begin{aligned} e_1 \cdot e_2 &= 0 \\ e_2 \cdot e_3 &= 0 \\ e_3 \cdot e_1 &= 0 \end{aligned} \quad (4.4)$$

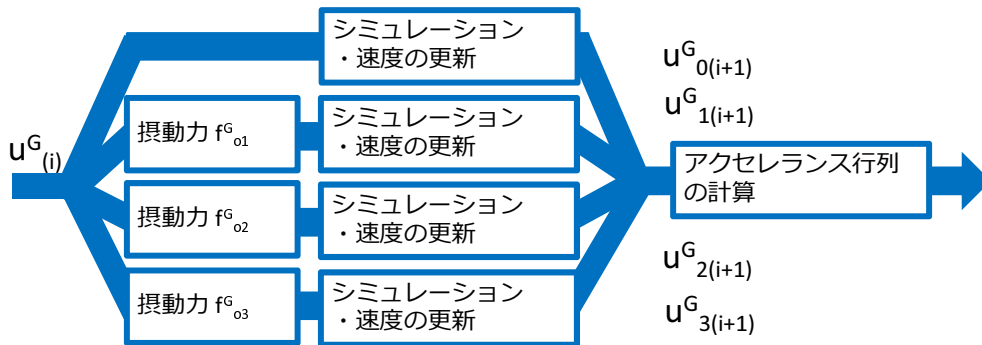


図 4.8: 摂動シミュレーションの手順

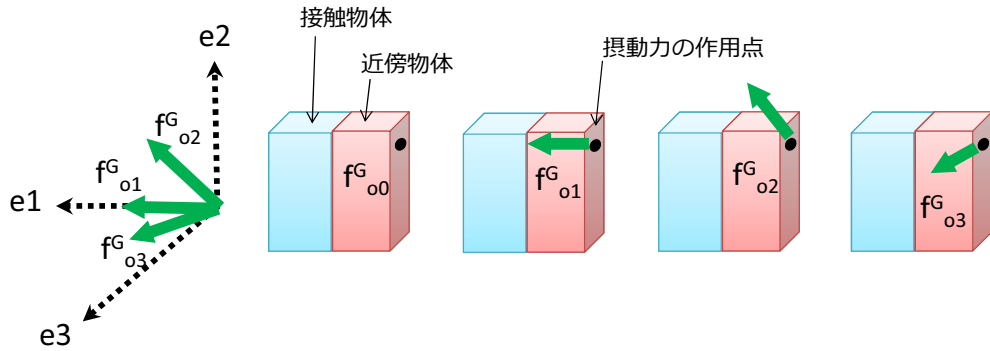


図 4.9: 摂動シミュレーションのための近傍物体に加える摂動力

次に得られた単位ベクトルと摂動力 f_{o1}^G のノルムから

$$f_{o2}^G = \frac{e_1 + e_2}{|e_1 + e_2|} |f_{o1}^G| \quad (4.5)$$

$$f_{o3}^G = \frac{e_1 + e_2}{|e_1 + e_2|} |f_{o1}^G| \quad (4.6)$$

のように2本の摂動力を求めることができる(図4.9)。求めた摂動力を

$$f_{o(i+1)}^G = \begin{cases} f_{o1}^G \\ f_{o2}^G \\ f_{o3}^G \end{cases} \quad (4.7)$$

と設定し、それぞれについて摂動シミュレーションを行い、1ステップ先の近傍物体の速度、角速度を求める。ユーザが実際に入力した力を使用する理由は、より正確な近傍物体のアクセラランスを求めるためである。 $f_{o(i+1)}^G$ のノルムが小さい場合には静止摩擦により、近傍物体は動かず、大きい場合には動摩擦により近傍物体は動く。

ここで摂動シミュレーションにより得られた近傍物体の速度、角速度を $u_{1(i+1)}^G$, $u_{2(i+1)}^G$, $u_{3(i+1)}^G$ とすると、これらはそれぞれ式(4.3)より、

$$\begin{cases} A_{(i+1)} f_{o1}^G = \frac{u_{1(i+1)}^G - u_{0(i)}^G}{\Delta t^G} + b_{(i+1)} (= y_1) \\ A_{(i+1)} f_{o2}^G = \frac{u_{2(i+1)}^G - u_{0(i)}^G}{\Delta t^G} + b_{(i+1)} (= y_2) \\ A_{(i+1)} f_{o3}^G = \frac{u_{3(i+1)}^G - u_{0(i)}^G}{\Delta t^G} + b_{(i+1)} (= y_3) \end{cases} \quad (4.8)$$

と書くことができる。さらに、式(4.8)をまとめると

$$A_{(i+1)} \begin{bmatrix} f_{o1}^G & f_{o2}^G & f_{o3}^G \end{bmatrix} = \begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix} \quad (4.9)$$

となる． $[f_{o1}^G \ f_{o2}^G \ f_{o3}^G]$ は 3×3 の正則行列なので逆行列が存在する．そこで，両辺右側から $[f_{o1}^G \ f_{o2}^G \ f_{o3}^G]^{-1}$ をかけると，

$$A_{(i+1)} = \begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix} \begin{bmatrix} f_{o1}^G & f_{o2}^G & f_{o3}^G \end{bmatrix}^{-1} \quad (4.10)$$

となることから，アクセラランス $A_{(i+1)}$ が得られる．得られたアクセラランス $A_{(i+1)}$ と重力等の外力に由来する加速度項 $b_{(i+1)}$ を第 $(i+1, 0)$ 回目の力覚スレッドステップに送信し，4.3.2 項で説明する局所シミュレーションにて使用する．

4.3 力覚スレッドの処理

4.3.1 3自由度力覚レンダリング

力覚ポイントの位置と近傍物体の情報からユーザへの提示力の計算を行う．提示力の計算はペナルティベースに従って行う．ペナルティベースは力覚ポイントが近傍物体に侵入した場合に，力覚ポイントと近傍物体を繋ぐ仮想のばねダンパモデルを用い，力覚ポイントが近傍物体に侵入した量とその微分に比例した値を提示力とする．

また，本研究では力覚レンダリングのために中間表現を採用している．この中間表現は物理スレッドと同期を行った際に更新される．この時，前回の中間表現は更新後の中間表現と必ずしも一致しないため，近傍物体を触り続けている場合に突発的な力を提示する必要がある．このような問題に対処するため，本研究では長谷川らが提案する中間表現の内挿 [35] を行う (図 4.10)．内挿を行い，中間表現を連続的に変化させることで，突発的な力の提示を防ぐ．

ここで，

- f_d : ユーザへの提示力
- k : ばね係数
- d : ダンパ係数
- x_p : 力覚ポイント側の接触点
- \dot{x}_p : 力覚ポイントの速度
- x_o^L : 近傍物体側の接触点
- \dot{x}_o^L : 近傍物体の速度
- n : 中間表現の面法線

として，図4.11のように力覚ポインタが近傍物体に侵入している時，ユーザへの提示力を式で表すと，

$$\mathbf{f}_{d(i,j)} = - \{ k(\mathbf{x}_{o(i,j)}^L - \mathbf{x}_{p(i,j)}) + d(\dot{\mathbf{x}}_{o(i,j)}^L - \dot{\mathbf{x}}_{p(i,j)}) \} \cdot \mathbf{n}_{(i,j)} \mathbf{n}_{(i,j)} \quad (4.11)$$

となる．また，近傍物体に加わる力 \mathbf{f}_o^L は作用反作用の法則より，

$$\mathbf{f}_{o(i,j)}^L = -\mathbf{f}_{d(i,j)} \quad (4.12)$$

となる．

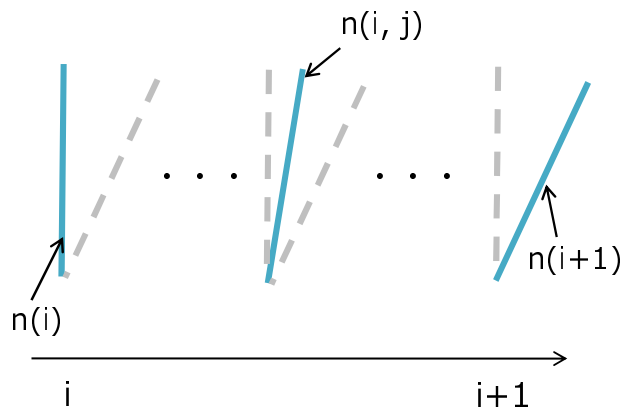


図 4.10: 中間表現の内挿

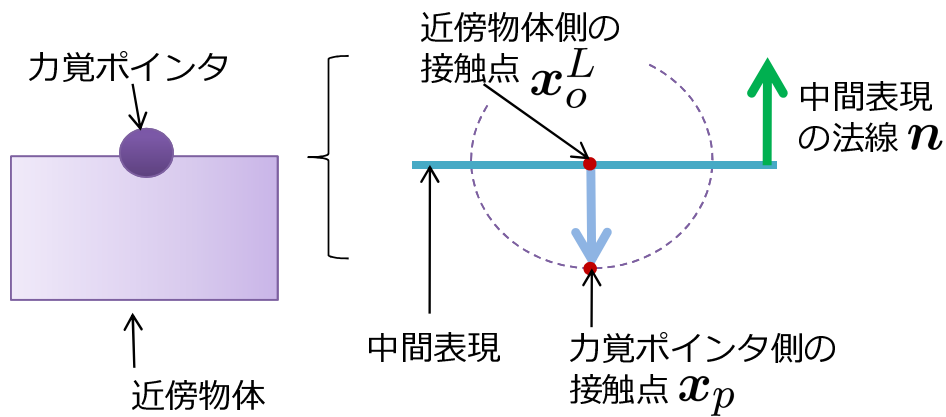


図 4.11:3 自由度力覚レンダリング

4.3.2 局所シミュレーション

前項で計算した力 f_o^L を用い、近傍物体について局所シミュレーションを行う。局所シミュレーションで使用する近傍物体の運動を表す式は4.2節で導出した近傍物体のアクセラランス A 、重力等の外力から由来する加速度項 b を使用して

$$\dot{u}^L = A f_o^L + b \quad (4.13)$$

となる。実際、第 $(i+1, j)$ 回目の力覚スレッドステップでの近傍物体の更新後の速度、角速度は式(4.13)の差分形

$$u_{(i+1,j)}^L = u_{(i+1,j-1)}^L + (A_{(i+1)} f_{o(i+1,j)}^L + b_{(i+1)}) \Delta t^L \quad (4.14)$$

として表すことができる。ただし、 Δt^L は局所シミュレーションの刻み時間である。また、更新後の近傍物体の位置、姿勢 s^L は

$$s_{(i+1,j)}^L = s_{(i+1,j-1)}^L + S u_{(i+1,j)}^L \Delta t^L \quad (4.15)$$

となる。ここで s は位置表現を3次元デカルト座標 x 、姿勢表現を四元数 $q = (q_w \ q_x \ q_y \ q_z)^T$ として $s = (x \ q)^T$ と表している。また、 S は角速度を四元数に変換する行列

$$S = \begin{pmatrix} E & 0 \\ 0 & Q \end{pmatrix}$$

で、 E は 3×3 の単位行列、 Q は

$$Q = \frac{1}{2} \begin{pmatrix} -q_x & -q_y & -q_z \\ q_w & -q_z & q_y \\ q_z & q_w & -q_x \\ -q_y & q_x & q_w \end{pmatrix}$$

である。

力覚スレッドステップを n 回行った後、近傍物体の位置、姿勢、速度、角速度と振動シミュレーションのための振動力 $f_{o(i+1,n-1)}^L$ を第 $i+2$ 回目の物理スレッドステップへ送信する。

4.4 大域，局所シミュレーションの同期処理

4.4.1 大域シミュレーションへ局所シミュレーション結果を反映

近傍物体の局所シミュレーション結果を大域シミュレーションに反映させる．反映は大域シミュレーションの近傍物体の位置，姿勢，速度，角速度を局所シミュレーションの近傍物体の位置，姿勢，速度，角速度に置き換える．実際，第 $i+2$ 回目の物理スレッドにおいて，第 $(i+1)$ 回目の力覚スレッドの局所シミュレーション結果を反映させると，

$$\mathbf{s}_{(i+2)}^G = \mathbf{s}_{(i+1,n-1)}^L \quad (4.16)$$

$$\mathbf{u}_{(i+2)}^G = \mathbf{u}_{(i+1,n-1)}^L \quad (4.17)$$

となる．これにより，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．

4.4.2 局所シミュレーションへ接触力以外の力の反映

近傍物体の局所シミュレーションは近傍物体に接触している物体からの接触力をアクセラランスとして考慮しているが，近傍物体と他物体の接触状況や静止摩擦，動摩擦の切り替わり等，線形的に変化しない力は考慮していない．これらの力を考慮しないと近傍物体に接触している物体が近傍物体に侵入する，近傍物体が受けた衝突力がユーザの手に伝わらない等の問題が起こる．そのため，大域，局所シミュレーション間での整合性を取る必要がある．

提案手法では大域シミュレーションで近傍物体に加わる非線形に変化する接触力以外の力 f_n^G を計算し，力覚スレッドを n 回行ううちの1回目に反映させ，大域，局所シミュレーションで扱う近傍物体の速度，角速度の整合性をとる．例えば，第 $i+1$ 回目の物理スレッドステップでの大域シミュレーションで近傍物体が衝突や摩擦状態の変化で力 $f_{n(i+1)}^G$ を受けた場合，第 $(i+2,0)$ 回目の力覚スレッドステップの局所シミュレーションへ力 $f_{n(i+1)}^G$ を反映させる．このとき第 $(i+2,0)$ 回目の力覚スレッドステップでの局所シミュレーションによる近傍物体の運動の更新は以下の式

に従って行う．

$$\begin{aligned} \mathbf{u}_{(i+2,0)}^L &= \mathbf{u}_{(i+1,n-1)}^L \\ &+ (\mathbf{A}_{(i+1)} \mathbf{f}_{o(i+2,0)}^L + \mathbf{b}_{(i+1)}) \Delta t^L \\ &+ \mathbf{M}^{-1} \mathbf{J}_{n(i+1)}^G \mathbf{f}_{n(i+1)}^G \Delta t^G \end{aligned} \quad (4.18)$$

ただし， M^{-1} は近傍物体の質量慣性行列 (6×6 行列)， J_n^G は力 f_n^G を力とトルクに変換するヤコビアン (6×3 行列) である．式 (4.18) の右辺第1項と第2項は4.3.2項で説明した式 (4.15) に相当し，第3項で近傍物体に接触力以外の力 $f_{n(i+1)}^G$ を加え速度，角速度を変化させる．このようにすることで，大域，局所シミュレーション間で整合性を取ることができる．

4.5 提案システムの評価

提案手法の有効性を確認するために，提案手法を実装した3自由度力覚提示システム使い，以下の評価を行った．

4.5.1 評価項目

- ・ 計算機シミュレーションによる力覚レンダリングの評価

提案手法により計算される提示力を他手法と比較し，提示力の正確さについて検証した．

- ・ 恒常法による主観評価実験

被験者実験により，提案手法と他手法の力覚提示に差があるかどうかを検証した．

- ・ 計算時間の評価

提案手法が他手法と比較して，どの程度計算に時間を要するのかを検証した．

4.5.2 実験環境

表 4.1 に実験を行うにあたり，構築した実験環境の詳細を示す．また，力覚提示の手法として下表 4.2 の 3 種類をシステムに実装した．シミュレーションの刻み時間は各手法の物理スレッドの更新周期と同値である．

表 4.1:ハードウェア構成

計算機	OS:Microsoft(R) Windows Xp Professional CPU:Intel(R) Core(TM)2 Duo 2.33GHz
力覚インタフェース	SPIDAR-G6[2]
物理シミュレータ	Springhead2[17]

表 4.2:システムに実装した手法

手法	物理スレッド更新周期	力覚スレッド更新周期
シミュレータを高速更新する手法	1ms	1ms
提案手法 (局所シミュレーションを行う手法)	50ms	1ms
従来手法 (シミュレータに力積を送信する手法 [35])	50ms	1ms

4.5.3 計算機シミュレーションによる力覚レンダリングの評価

提案手法が従来手法と比較して正確に力覚レンダリングができているのかを評価するための計算機シミュレーションを行った．

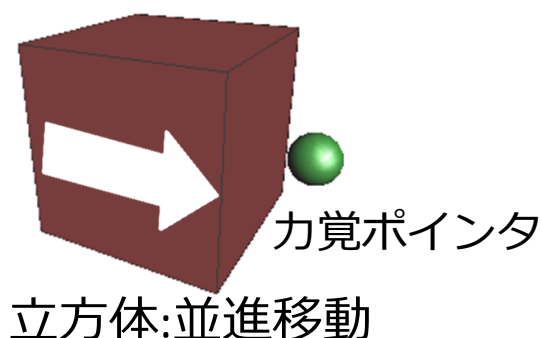


図 4.12:構築したバーチャル世界

シミュレーション内容

図 4.12 のような重力のない 3 次元バーチャル世界上で立方体を並進移動させる。この立方体を球形状の力覚ポインタに衝突させ、はね返る間に力覚ポインタが立方体に加える力を計測する。力覚レンダリングによる力を全て立方体に加えるために、力覚ポインタの位置更新、速度更新はせず、その場に留まるように設定した。また、立方体の回転運動が力覚レンダリングに影響することを考慮し、立方体は回転運動をさせず、力が立方体の面に対して垂直に加わるようにした。

上記のシミュレーションを実装した 3 手法 (シミュレータを高速更新する手法、提案手法、従来手法) について立方体の質量 60kg、速度 0.017m/s で行った。

シミュレーション結果

図 4.13 にシミュレーション結果を示す。図 4.13 から従来手法はシミュレータを高速に更新する手法に比べて、立方体と力覚ポインタとの接触時間が長く、立方体へ加える力が大きくなっている。一方、提案手法は立方体と力覚ポインタとの接触時間と力がシミュレータを高速に更新する手法と同様の軌跡が得られている。

次に立方体と力覚ポインタが接触している間に力覚ポインタが立方体に加えた力積を表 4.3 に示す。力積の値から、従来手法は他の手法に比べて大きな力積を立方体に加えており、提案手法はシミュレータを高速更新する手法と同値であった。このことから提案手法は従来に比べてより正確に力覚レンダリングができていると考えられる。

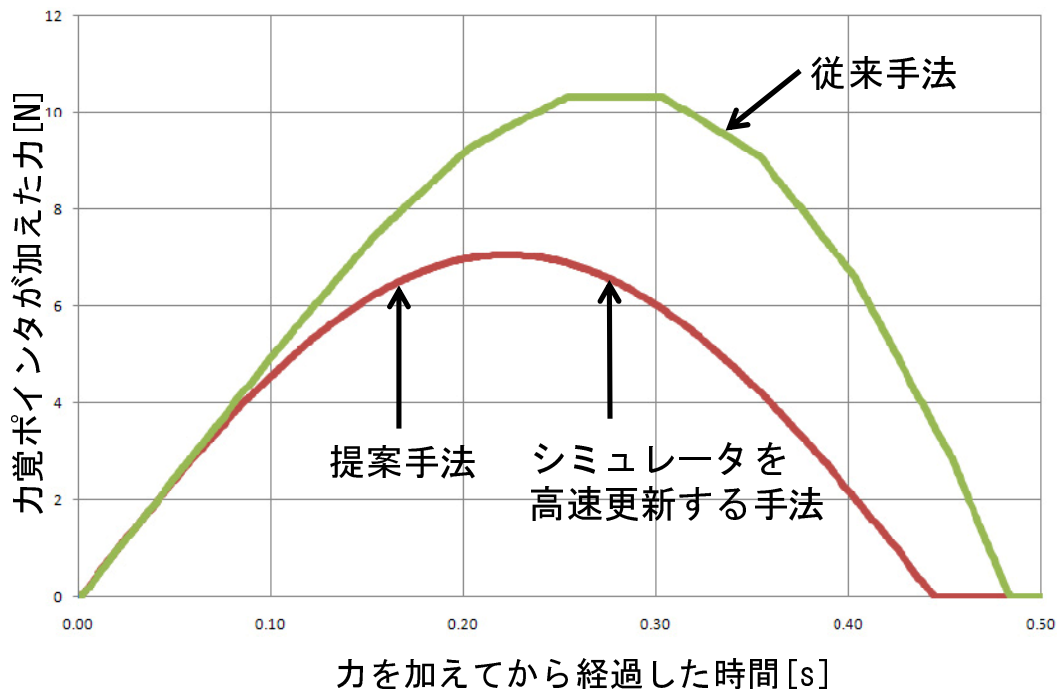


図 4.13:力覚レンダリングによる力

表 4.3:力覚ポイントが立方体に加えた力積

	力積 [Ns]
シミュレータを高速更新する手法	2.0
提案手法	2.0
従来手法	3.2

4.5.4 恒常法による主観評価実験

4.5.3 項の計算機シミュレーションで従来手法が提案手法よりも大きな力を提示していたことを確認した。本項では被験者に力覚提示を行い、提案手法の効果を評価する。

実験内容

4.5.3 項で構築した重力のないバーチャル世界上で並進移動する立方体 (図 4.12) を、被験者に力覚インタフェースを用いて受け止めてもらった。被験者は図 4.14 のよう



図 4.14: 実験の様子

にクッションに肘を置き，力覚提示による腕の動きを制限させた．また，バーチャル世界をCG(コンピュータグラフィックス)として被験者の視覚に提示することにより，立方体と力覚ポインタとの衝突のタイミングを被験者が予測してしまう可能性がある．それにより，被験者が本能的に手を動かしてしまい，力覚提示に影響がでることが考えられる．そこで視覚提示による力覚提示への影響を考慮して，被験者へはバーチャル世界のCGを提示しなかった．

本実験では1試行で標準刺激，比較刺激の質量いずれかをそれぞれ1回ずつ順序をランダムにして提示した．提示する刺激は標準刺激，比較刺激ともに一定速度で被験者の正面左から右に移動し，被験者は移動する立方体を力覚ポインタで受け止めさせた．その後，被験者にどちらが重いかを2件法により回答させた．被験者への提示刺激を以下に示す．

標準刺激

- ・ シミュレータを高速更新する手法
 - 質量 60kg
 - 速度 0.017m/s

比較刺激

- ・ 提案手法 (局所シミュレーションを行う手法)
- ・ 従来手法 (シミュレータに力積を送信する手法)

- 質量 30, 45, 60, 75, 90, 105kg
- 速度 0.017m/s

1種類の比較刺激のうちの1種類の質量について20試行を行い、1種類の比較刺激で計120試行を行った。

実験結果

実験の結果として、被験者(A~E)が比較刺激のほうが重いと回答した回数を図4.15に示す。

次に、実験結果のデータに対して最尤度法によるロジスティック回帰曲線関数を用い、カーブフィッティングを施し、主観的等価点を求めた。求めた主観的等価点を表4.4に示す。

表4.4:主観的等価点

被験者	提案手法 [kg]	従来手法 [kg]
A	57.4	21.3
B	68.6	23.3
C	53.7	37.1
D	58.4	33.0
E	58.1	43.7
平均	59.2	31.7

主観的等価点についても同様に被験者全員に対して、従来手法のほうが提案手法に比べて立方体が重いと感じていることがわかる。標準刺激60kgと各手法の主観的等価点の平均値についてみると標準刺激と提案手法との差は-0.8kg、標準刺激と従来手法との差は-28.3kgである。

さらに、得られた結果に対して、検定を永井らが提案する手法[50]を用い検定を行った。検定はつぎのように行った。まず、主観的等価点を求めるためのカーブフィッティングの際にロジスティック回帰曲線のフィッティングパラメータ θ_1 、 θ_2 の分散と共分散を求めた。そして、フィッティングパラメータ θ_1 、 θ_2 から主観的等価点の分散を求め、 χ^2 値を計算し、1自由度 χ^2 分布図から p 値を計算した。その結果、

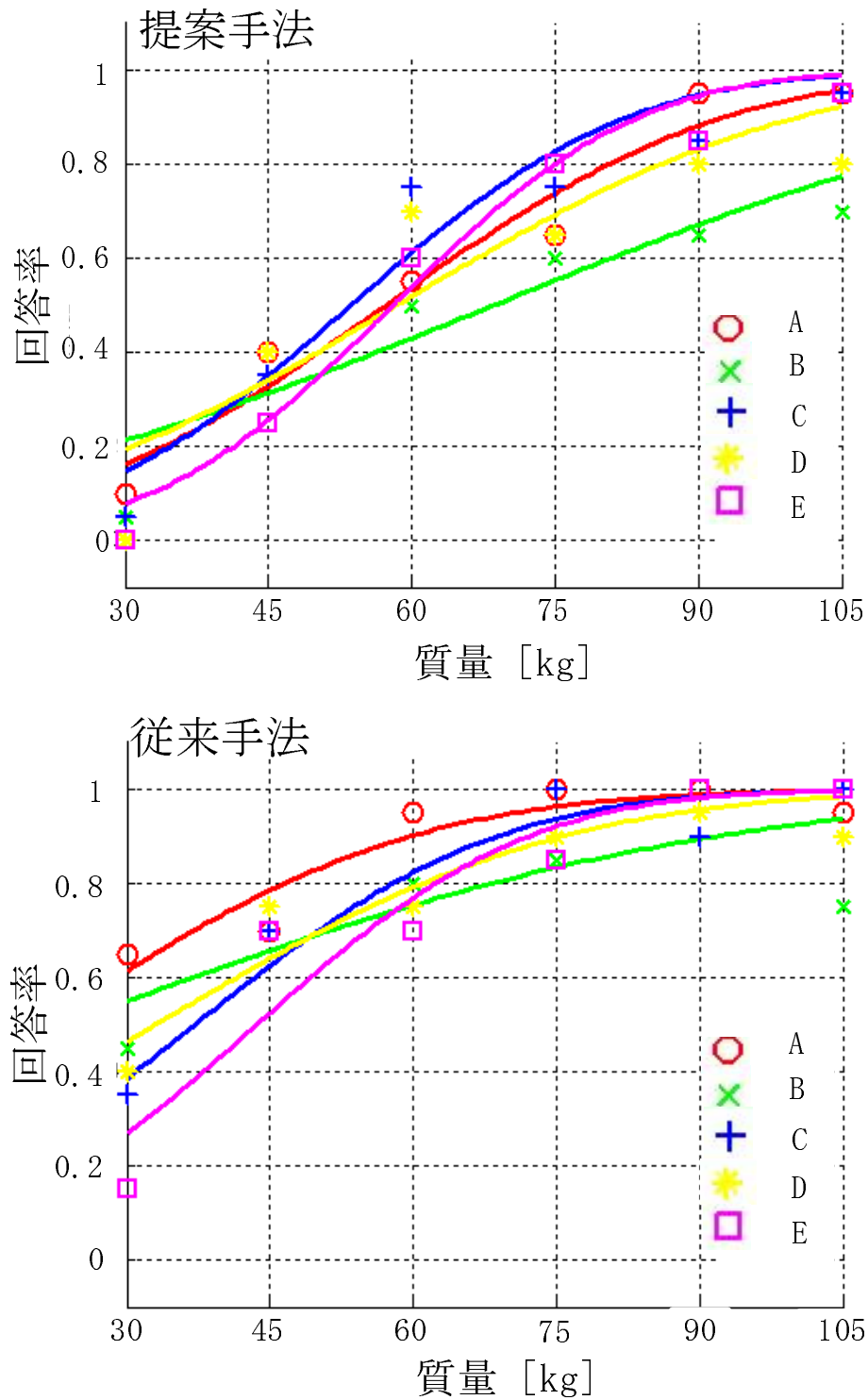


図 4.15: 恒常法による被験者実験結果

$p = 0.45$ となった．これは有意水準 0.05 を上回っているため，有意差はないと判定した．

4.5.5 計算時間の評価

提案手法では通常の物理シミュレーションに加え，摂動シミュレーションを行っている．そこで，摂動シミュレーションがどの程度計算時間を増加させているか調べるために，シミュレーションの1ステップ当たりの計算時間を測定した．

実験内容

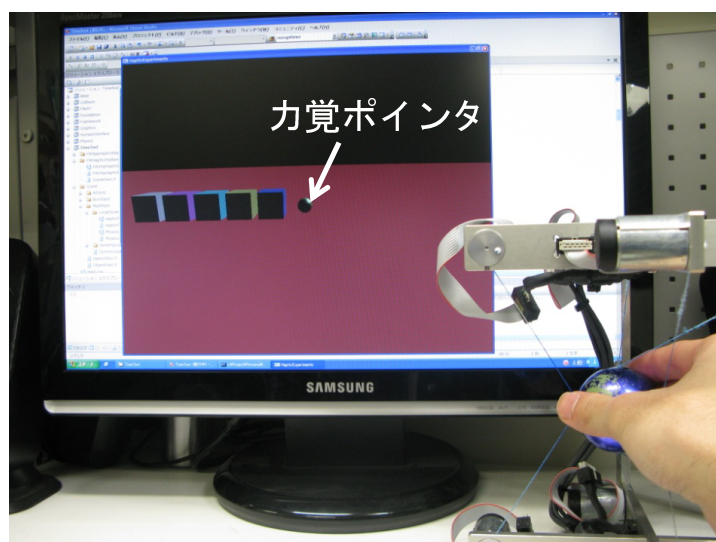


図 4.16: 近傍物体に接触する物体の影響

図 4.16 のような床の上にある立方体を力覚ポインタで押している時の物理シミュレーション1ステップ当たりの計算時間を計算機のCPUクロック時間を利用して計測した．計測はシミュレータを高速更新する手法，提案手法，従来手法で行い，提案手法については摂動シミュレーションの計算にかかる時間も合わせて計測した．また，物理シミュレーションされる物体の個数によって計算時間が変化するので，各手法について立方体の個数を1，10個にして計測した．個数が10個の場合は始めは立方体同士は接触しておらず，力覚ポインタで押していくうちに接触するようにした．

計測結果

計測結果を図 4.17, 図 4.19 に示す。これらのグラフの縦軸は物理シミュレーション 1 ステップの計算時間を刻み時間で除算した割合で、刻み時間当たりの計算占有率である。計算占有率は刻み時間内に計算が完了しているかどうかをみる指標で、計算占有率が 1 を下回る場合は刻み時間内に計算が完了することを意味し、1 を超える場合は刻み時間内に計算が完了しないことを意味する。

まず、図 4.17, 図 4.19 から物体の数の違いによる計算占有率の変化に着目してみると、各手法について立方体の数が多いほうが物理シミュレーション 1 ステップの計算占有率が増加している。シミュレータを高速更新する手法では立方体の数が 1 個の場合は計算占有率は 0.15 を下回りリアルタイムに計算できているが、立方体の数が 10 個となると計算占有率が 1 を超えるようになり、設定した刻み時間 1ms 以内に計算が終わらず、刻み時間の約 3 倍の計算時間を要している。実際に時間計測中の力覚インタフェースを用いた力覚提示も発振が起きるようになり不安定であった。提案手法、従来手法については立方体 10 個の場合についても計算占有率が 0.25 を下回り、リアルタイムに計算を終えている。力覚インタフェースを用いた力覚提示も発振は起きず、安定に力を提示できていた。

次に、提案手法と従来手法の結果を拡大したもの(図 4.18, 4.20)についてみる。参考のため、提案手法の結果の内摂動シミュレーションを含まない大域シミュレーションのみを測定したものを破線で示す。提案手法(大域シミュレーションのみ)と従来手法では立方体の数の変化で計算占有率に大きな差は見られなかった。提案手法(摂動シミュレーションを含む)と従来手法を比較すると提案手法(摂動シミュレーションを含む)は従来手法に比べ、約 3 倍から 4 倍となっている。これは、摂動シミュレーションで力覚ポインタが力を加えない場合とそれぞれ 3 方向に力を加える場合の計 4 回のシミュレーションを行い、接触する立方体の個数に応じて拘束力の計算をしているため、計算時間が増加しているといえる。

以上のように、提案手法はシミュレータを高速更新する手法に比べて多くの物体をリアルタイムにシミュレーションすることができている。また、従来手法に比べて、4.5.3 項での実験の結果から正確に力覚提示ができているが、摂動シミュレーションの分だけ計算時間が増加していることがわかった。

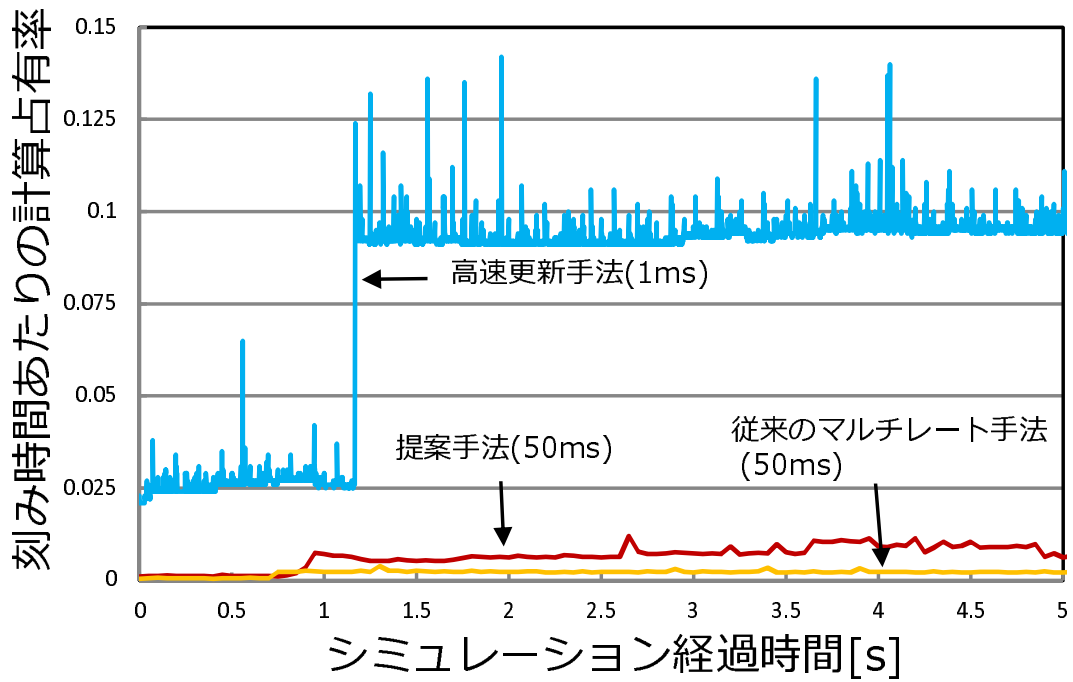


図 4.17:立方体数 1 での計算時間

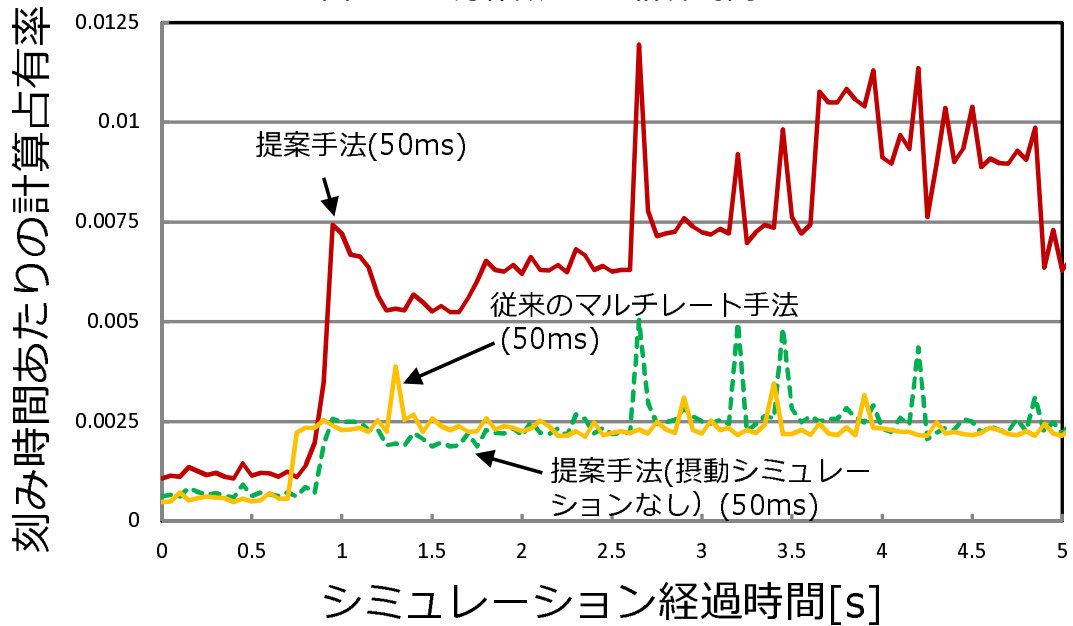


図 4.18:立方体数 1 での計算時間 拡大図

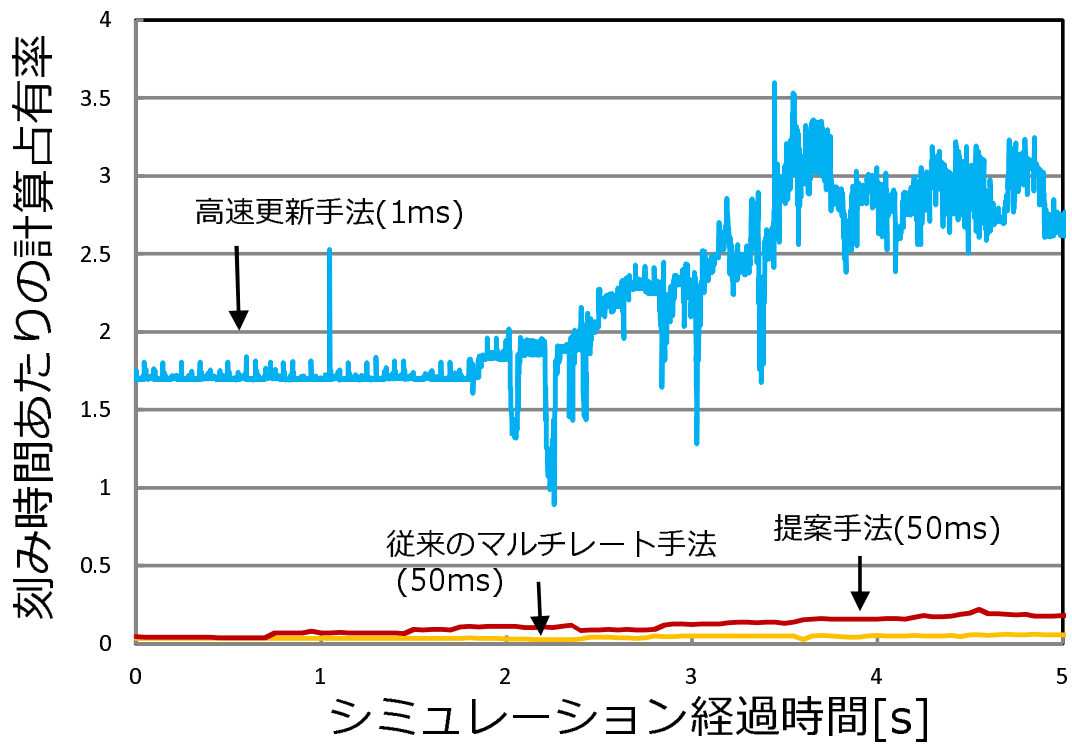


図 4.19:立方体数 10 での計算時間

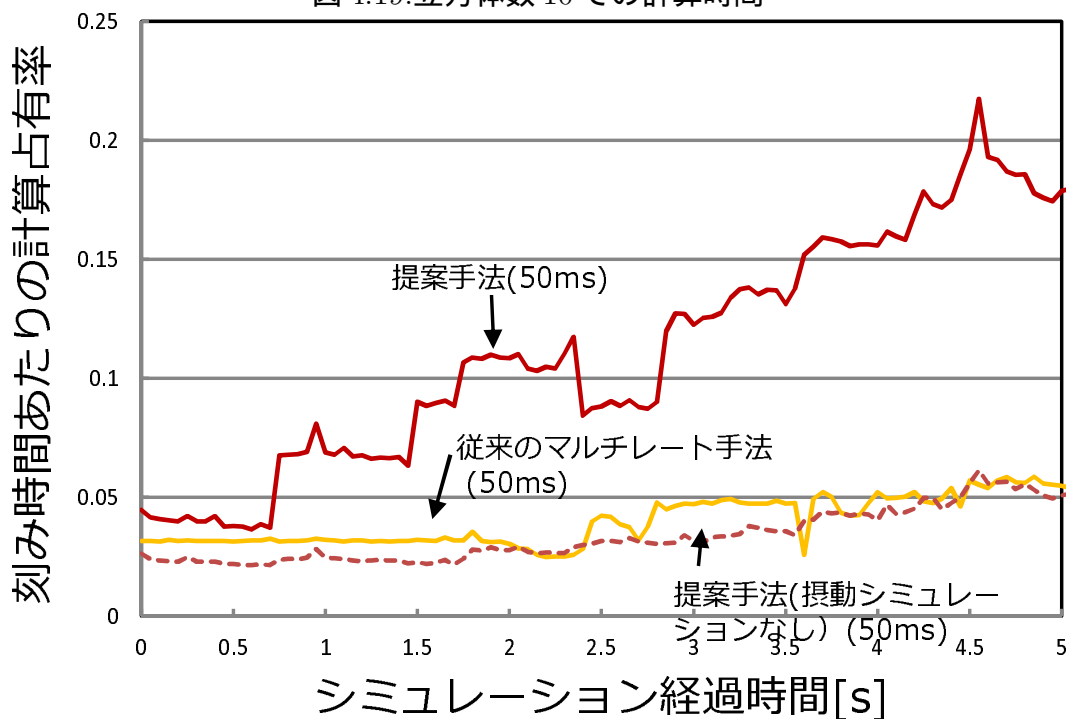


図 4.20:立方体数 10 での計算時間 拡大図

4.6 考察

4.6.1 近傍物体のアクセラランスの効果

近傍物体に接触している物体の接触力が近傍物体のアクセラランスで考慮されているかを確認するために、図 4.16 のような状況で力覚ポインタを使い立方体を押し、提示される力の変化を確かめた。結果、立方体同士が接触するごとに手にかかる力が大きくなっていることを感じ、近傍物体のアクセラランスに近傍物体に接触している物体からの接触力が含まれていることが確認できた。

また、4.5.3 項の実験では近傍物体の運動を並進移動に制限しており、近傍物体の回転運動について評価はしていない。しかし、力覚インタフェースを用いて立方体の形状をした近傍物体の辺に近い部分に力を加えると、違和感なく近傍物体が回転していることを確認した。

4.6.2 提案手法の適用範囲

本章では提案手法の適用範囲を剛体に絞って議論を進めた。有限要素解析が必要な弾塑性体については検討していない。提案手法を弾性体に適応させるには更に検討が必要だと考えられる。ただし、剛体を関節で繋ぐことで、鎖や針金のような物体とのインタラクションは可能である（図 4.24）。

また、力覚ポインタと剛体の間は点接触している、力覚ポインタが剛体を押すことを考え、3 自由度の力覚インタラクションについて議論した。力覚ポインタと剛体間で多様な接触を実現し、応答性のあるインタラクションを実現するためには、3 章で提案している 6 自由度の力覚レンダリングを行い、6 自由度の入力に対応したアクセランス行列を求める必要がある。

加えて、提案手法は近傍物体の数だけ高速に局所シミュレーションを行う。そのため、近傍物体の数が増えると計算量が増加し、実時間での計算が難しくなる。例えば図 4.21.(a) のように、力覚ポインタについて近傍物体が 1 つであれば、近傍物体 1 つを局所シミュレーションすれば良い。しかし、図 4.21.(b) のように、力覚ポインタについて近傍物体が多数あると、全ての近傍物体を局所シミュレーションする必要があり、計算が実時間で完了しなくなる。今回実装したシステムでは数個（8 個程度）の近傍物体を実時間でインタラクション可能であったが、それ以上は更新

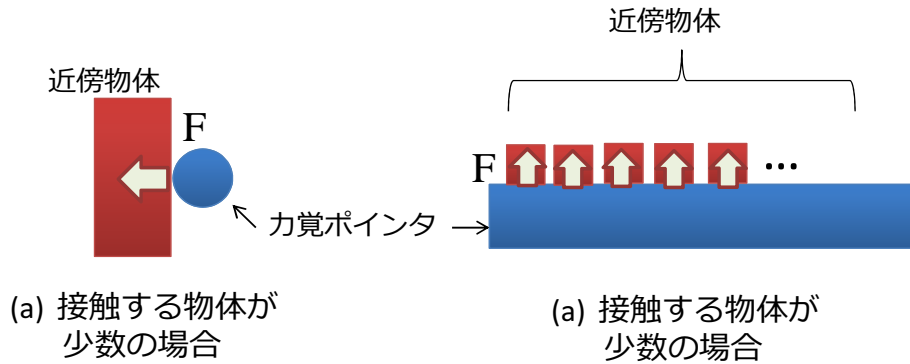


図 4.21:近傍物体の数の限界

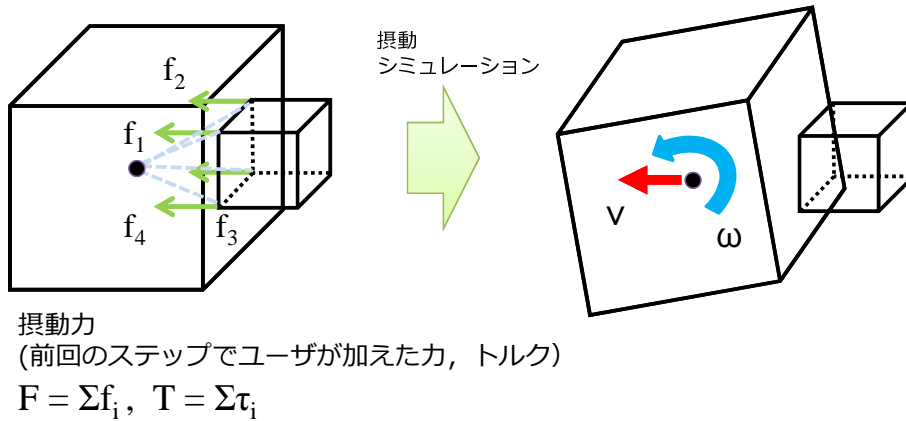


図 4.22:6 自由度力覚インタラクションのための摂動シミュレーションの検討

速度が低下していた．そのため，多数の物体とのインタラクションを実現したい場合は，並列化を行うなど計算量を分散させる必要がある．

4.6.3 6 自由度力覚インタラクションへの拡張の検討

前項で述べたように，局所シミュレーションと6自由度力覚インタラクションを実現するためには，並進力，トルクを摂動力として近傍物体に加え，シミュレーションを行い速度変化を知る必要がある．そこで，図 4.22 のように無重力状態で力覚ポインタが近傍物体に面（4点）で接触し，押す状況を考える．力覚ポインタと近傍物体の各接触点に対して，力が働き，合力は $F = \sum f_i$ となる．さらに，近傍物体には加えられた力に対してトルクが加わり，合トルクは $T = \sum \tau_i$ となる．ここで，各接

触点から算出されるトルクがうまく打ち消し合えば $T = (0\ 0\ 0)^t$ となり、トルクは働かず、角速度は変化しない。一方、各接触点から算出されるトルクが打ち消し合わなければ $T \neq (0\ 0\ 0)^t$ となり、トルクが働き、角速度が変化する。つまり、 τ_i の向き、大きさにより、近傍物体の回転運動が変化する。以上のようにトルクによって、近傍物体の回転運動が変化することから、いくつかの方向に対して摂動トルクを与え、近傍物体の運動を観察する必要がある。そのため、いくつかのパターンの摂動トルクが必要になるのかを考える。

力覚ポイントと近傍物体の接触が1点のみの時は、加えられる力に対して、一意にトルクが決まる。よって、3方向の摂動力を加える場合と何も力を加えない場合について、それぞれ摂動シミュレーションを行えば良いので、4回摂動シミュレーションを行えば良い。一方、接触が多点の時は、各接触点の侵入量の大きさによってトルクの変化する。また、トルクは1自由度に対して正方向、負方向の2通りがある。よって、1方向の摂動力に対して、6方向の摂動トルクを組み合わせ、摂動シミュレーションを行う必要がある。つまり、3方向の摂動力になると摂動力を加えない場合を合わせて、計19回の摂動シミュレーションが必要となる。ここで、4.5.5項で行った計算時間の評価では、立方体を10個押した場合、提案手法は計算占有率が最大で約0.22であった。これは、摂動シミュレーション4回と大域シミュレーション1回、計5回のシミュレーションを行った時の計算占有率である。一方、6自由度のインタラクションを実現するには、摂動シミュレーション19回と大域シミュレーション1回、計20回のシミュレーションが必要となり、計算占有率を図4.19から概算すると、約1.0となる。従って、実時間で計算が完了するのは立方体10個程度までであるといえる。以上のように、今回提案する方法で6自由度のインタラクションを実現するためには多くの計算量を必要とし、大規模なバーチャル世界上で実現することが難しい。そのため、局所シミュレーションを用いた力覚提示システムを利用し、6自由度の力覚インタラクションを実現するには別の方法を検討する必要がある。例えば、力を加えない場合と力を1回だけ加える場合で摂動シミュレーションを2回行い、その結果からアクセラランスを推定する方法が考えられる。この方法は推定に基づくため、精度は実際に検証を行わないとわからないが、計算量を削減する方法としては有効であると考えられる。

4.6.4 実装したシステムを利用したインタラクションの例

今回、実装したシステムを利用して、多数の多面体（図 4.23）や多関節で繋がれた剛体（図 4.24）とインタラクション可能なデモを作成した。多数の多面体のデモでは 80 個程度の立方体、球体、多面体、カプセルとインタラクション可能で、それぞれの運動、形状、重さが力覚提示され体感することができる。多関節で繋がれた剛体のデモは電気スタンドの首の曲げ操作を行うことができ、光源の向きを操作することができる。曲げの抵抗を感じながら操作を行うことができるので、直感的に所望の向きに光源を向けることができる。以上のように、提案手法により大規模なバーチャル世界とのインタラクションが実現できていることがわかる。

また、提案システムでは幾何学的な中間表現 [32] を利用した力覚レンダリングを行なっているため、力覚ポインタが複数の物体と接触している場合でも力覚ポインタと各剛体間に働く力を知ることができる。そのため、力覚ポインタが各剛体に接触した瞬間に減衰振動を提示力に付加することで、アルミや木材といった物体固有の振動を提示することができる [51][52]。提案システムにおいて、各材質を叩いた時に生じる振動を力覚として付加することで、バーチャルカップリング法だけでは再現できない異なる材質を提示することができた（図 4.25）。このように、提案システムでは高品質な力覚提示が実現できているといえる。

4.7 第4章のまとめ

従来ではスレッド分割された力覚提示システムにおいて、スレッド間での通信遅延のため正確に力覚提示を行うことができなかった。本章では動力学の力覚提示に必要な情報、アクセラランスを明らかにすることで、力覚スレッド内でバーチャル世界の一部を高速にシミュレーションすることが可能な局所シミュレーションによる手法を提案した。そして、正確に力覚提示が実現できているか評価するためのシミュレーションを行い、提案手法が従来手法に比べて有効であることを確認した。また、各手法の計算時間を測定し、シミュレータを高速更新する手法に比べて提案手法は大規模なバーチャル世界上で力覚提示が可能であることを確認した。これらの評価から提案手法によりスレッド分割された力覚提示システムにおいても正確に力覚提示を実現できた。

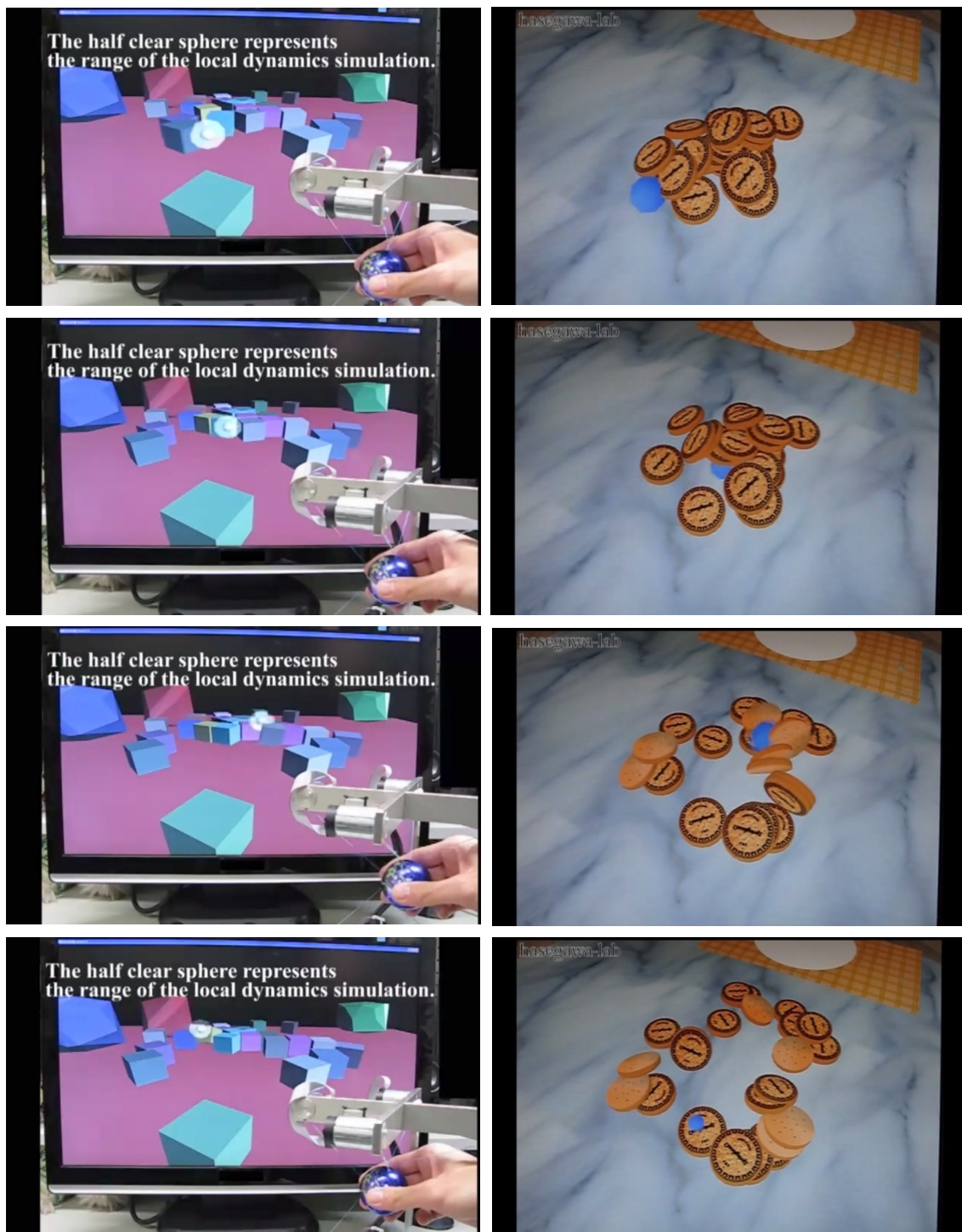


図 4.23:多数の多面体との力覚インタラクション

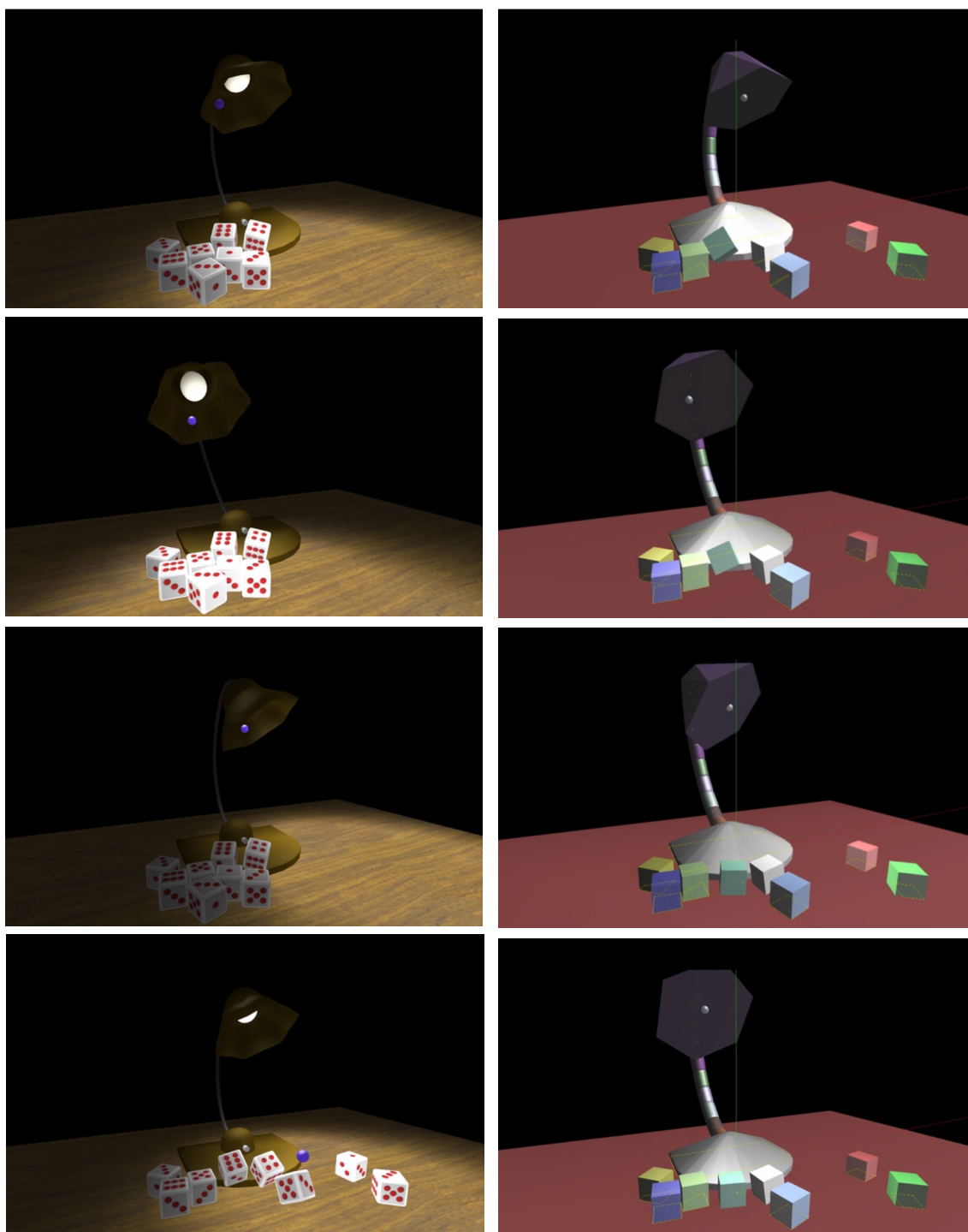
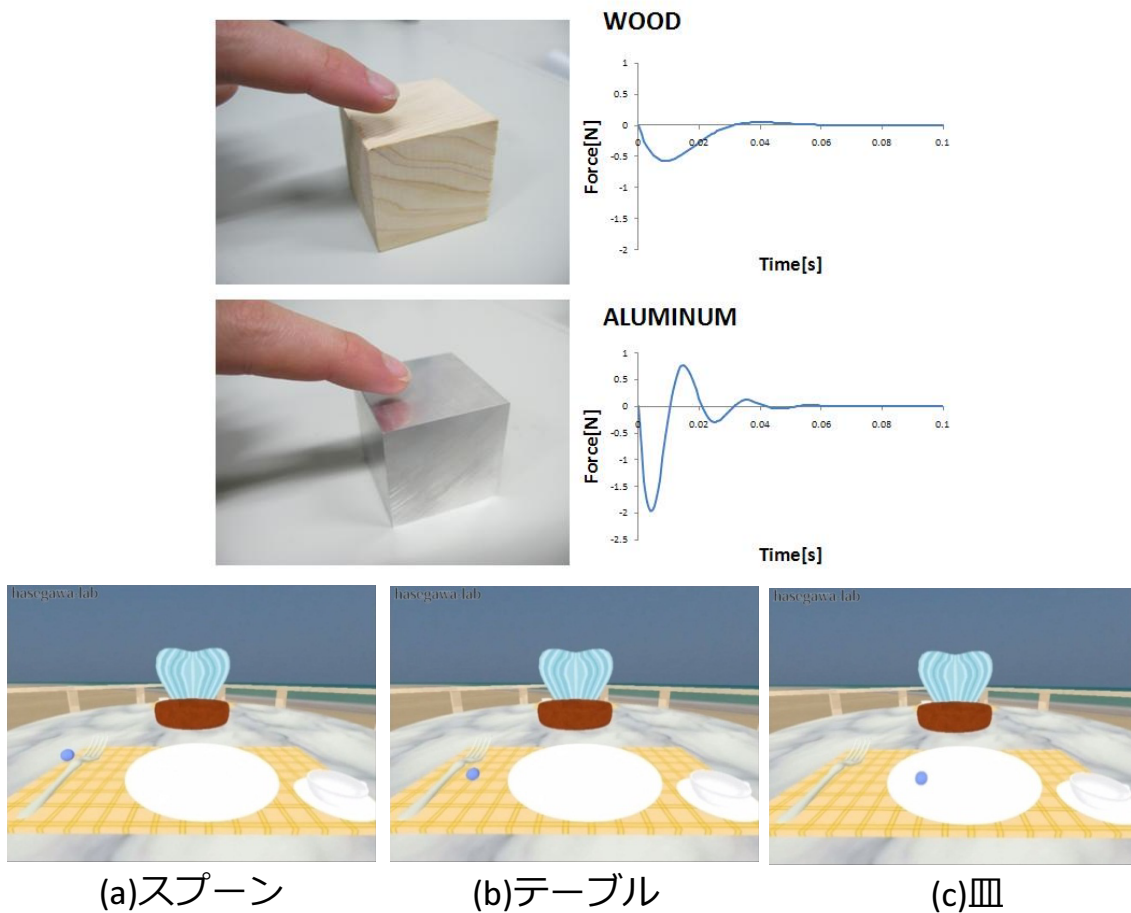


図 4.24:多関節剛体とのインタラクション



それぞれ異なる振動を提示することで、材質感を表現

図 4.25:固有振動の付加による材質感の提示

第5章

結論

5.1 本論文のまとめ

本論文では6自由度の力覚提示に必要な情報と動力学の力覚提示に必要な情報を明らかにし、それぞれの情報を利用した力覚提示システムの提案をした。そして、高品質な力覚インタラクションを実現するマルチレート力覚提示システムを実現した。

第1章「序論」ではバーチャル世界との力覚インタラクションについて述べ、力覚インタラクションが設計、教育、訓練、エンタテインメント等の分野において応用されていることを述べた。また、力覚提示システム、特に剛体のバーチャル世界との力覚インタラクションを実現するために必要な要素である物理シミュレーション、力覚インタフェース、力覚レンダリングを挙げ、それぞれについて説明を行った。そして、これらの要素により構成される従来の力覚提示システムについて触れ、特徴、問題点について述べ、高品質な力覚インタラクションのための力覚提示システムの実現することを本研究の目的とした。目的の実現のために、6自由度の力覚提示に必要な情報と動力学の力覚提示に必要な情報を明らかにし、それぞれの情報を利用した力覚提示システムの提案をすることを述べた。

第2章「マルチレート力覚提示システム」では大規模なバーチャル世界との力覚インタラクションを実現する方法として従来から提案されているマルチレートシステムの研究を紹介した。さらに、従来から提案されている手法について問題点と原因を考察し、高品質な力覚インタラクションを実現するために必要な事項を検討した。そして、以下の項目が課題であると述べた。

- ・ 不要な慣性力のでない6自由度の力覚提示
- ・ 規模の大きいバーチャル世界
- ・ 時間遅れのない応答、整合性

第3章「6自由度力覚レンダリングのための中間表現」では6自由度力覚レンダリングに必要な情報である幾何学的な中間表現を明らかにし、マルチレートな6自由度力覚提示システムを提案した。6自由度の力覚レンダリングに必要な情報を明らかにすることで、大規模なバーチャル世界に対して不要な慣性力が入らない6自由度の力覚インタラクションを実現できる。6自由度の力覚レンダリングに必要な情報は、力覚ポインタ（ユーザがバーチャル世界内で操作する物体）がバーチャル物体と接触した時の、力覚ポインタの接触点群とバーチャル物体に載る平面（力覚ポインタとバーチャル物体間の抗力方向を法線とする平面）である。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評価を行い、6自由度の力覚提示、摩擦力の提示が適切に行え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確認できた。

第4章「局所シミュレーションを用いた力覚提示システム」では動力学の力覚提示に必要な情報を明らかにし、ユーザの入力に対してシステムの応答に遅れのない正確な力覚提示が可能なシステムを実現する。動力学の力覚提示に必要な情報とは力と加速度の比例係数であるアクセラランスと加速度項で動力学的な中間表現といえる。アクセラランスを利用すると、力覚ポインタ近傍の物体の運動を高速にシミュレート（局所シミュレーション）することができ、ユーザが加える力を即座に反映させることができる。さらに、近傍物体以外の物体の運動は低速にシミュレート（大域シミュレーション）するので、計算負荷を抑えることができる。評価としてシミュレーションを行い、提案手法が従来のマルチレートシステムに比べて正確な力覚レンダリングが行えていることを確認した。また、各手法の計算時間を測定し、シミュレータを高速更新する手法に比べて提案手法は大規模なバーチャル世界上で力覚提示が可能であることを確認した。最後に本手法の適用範囲は3自由度の力覚インタラクションまでであるため、6自由度の力覚インタラクションへ拡張させるための検討を行った。

以上のように、本論文では高品質な力覚インタラクションを実現するためのマルチレート力覚提示システムの提案をした。本研究により、実世界とバーチャル世界の境界を薄め、現実と見紛うほどの力覚インタラクションの実現に近づいていると

言える。

5.2 応用と展望

5.2.1 応用

本論文で提案した手法は様々な分野に応用できると考えられる。例えば第3章「6自由度力覚レンダリングのための中間表現」で提案した手法は不要な慣性力が提示されることがない。そのため、力覚ポインタを手の一部として捉えることができ、力覚インタフェースの存在を意識せずにバーチャル世界とインタラクションすることができるので、複雑な操作が要求される設計、組立等の作業支援や体感型のエンタテインメントシステムへの応用が考えられる。

第4章「局所シミュレーションを用いた力覚提示システム」は大規模なバーチャル世界に対して応答性のあるインタラクションを提供するものである。応答性のあるインタラクションは正確な力覚を提示することにつながり、例えば質量が異なる物体を正確に操作するといった場合に効果があると考えられる。つまり、上記で挙げた組立操作や手術のシミュレーションなど力の微妙な変化が要求される操作への利用が考えられる。また、計算量を分散するための方法として、並列計算を行うことが考えられる。提案手法では動力学的な中間表現（アクセラランス、加速度項）を求めるために4回の摂動シミュレーションを行う。これらは、独立して実行可能なため、それぞれ並列計算が可能である。つまり計算時間は大域シミュレーション1回と摂動シミュレーション1回分で済むので、今回実装したシステムよりも2.5倍高速に計算できる可能性がある。また、それぞれのシミュレーションの中では衝突判定、拘束力の計算・積分が行われているが、それぞれについても計算を分散させることができる。衝突判定では衝突する物体の各ペアについて接触点、抗力の方向を求める部分があり、各ペア毎に並列化が可能である。拘束力計算・積分は拘束条件から抗力を計算し、積分をして物体の運動を更新しており、拘束条件が独立している部分については、並列化できる。拘束力計算・積分の並列化によるシミュレーションの高速化はシーンの状態によって大きく変わるが、衝突判定の並列化は効果的であると考えられる。

また、4.6.3項で述べたように局所シミュレーションを用いた力覚提示システムを6自由度の力覚インタラクションに拡張させることで多様な接触状態と正確な

力覚提示を実現することができ、大規模なバーチャル世界に対してより高品質な力覚インタラクションが可能となる。これにより、人間が実世界に対して行うインタラクション、道具を利用して物を動かす、組み立てるといったことがバーチャル世界でも可能となる。さらに、装着型の力覚インタフェース、例えばSPIDAR-8[26]やSPIDAR-MF[53]、Hiro[3]と組み合わせれば、多指によるインタラクションが可能となり、バーチャル世界の中でも把持操作を行うことができる。また、多指によるインタラクションは複数の力覚ポイントが指に配置されることになるので、複数の力覚ポイントが利用できるといえる。複数の力覚ポイントが利用できるということは、一人のユーザだけでなく複数のユーザが利用できるようになるので、[54][55]が提案しているような複数人での協同作業が行えるようになると思われる。

5.2.2 展望

今後の展望として、次のことが挙げられる。まず、本論文での提案手法の適用範囲は剛体モデルに限られているため、組立操作やロボットとのインタラクションなど応用範囲が限られる。そこで、提案手法を柔軟物体や流体シミュレーションへ拡張し、より多くの力学系への適用が考えられる。そして、各シミュレーションを連成させることで、これまでに提案されている手術[56]や調理[57]の訓練、バーチャル化学実験といった工学、医療分野で必要とされるアプリケーションへ応用でき、従来よりも高品質なインタラクションが可能になることが期待できる。提案手法をより多くの力学系への適用にさせるには、その力学系に合わせた中間表現を導出し利用することになる。特にメッシュで表現される柔軟物体は[58]で提案されているように、三角形メッシュと剛体の頂点の時間変化による幾何学的な関係を連続衝突判定とカーブフィッティングにより求めることで、正確な反力を計算し、運動を求めることができている。本研究で提案した動力的な中間表現は力と運動の関係を表しているため、上記のようなシミュレーションにより、柔軟物体にも適用できると考えられる。

また、バーチャル世界にバーチャルクリーチャー[59]を出すことで、バーチャルクリーチャーとの物理的なインタラクションを実現することができる。バーチャルクリーチャーは無機的なバーチャル世界を有機的で魅力あるものに変える。そして、ユーザとバーチャルクリーチャー間で物理的なインタラクションができると、ユーザにとってバーチャルクリーチャーの存在感が増し、バーチャル世界への没入感を

高められることが期待できる。これにより、ビデオゲームやメディアアートといったエンタテインメント分野やバーチャルペットとして福祉の分野で利用できると考えられる。

さらに、学術的な展望も考えられる。現在、力触覚の分野では人間の力触覚の知覚特性を調べ、それに合わせて力触覚を提示する研究が盛んに行われている [60]。刺激（力覚，圧覚，すべり触，振動覚等）がどういった感覚に対応するのかを調べることで、人間の力触覚の解明につながる。本研究で提案したシステムは、力覚レンダリングの方法の変更が容易であることから様々な感覚を提示することができる。例えば 4.6.4 項で述べたように、振動を付加することで材質感，硬さの提示が可能で、人間の材質知覚の解明につながることを期待できる。

謝辞

本研究を行う機会を頂き，ご指導頂いた長谷川晶一准教授，佐藤誠教授に感謝いたします。力覚インタラクションの研究に従事することができ，充実した大学生活を送ることができました。また，研究以外のことについても，多くのことを学び見聞を広めることができました。研究を進めるに当たり，ご助言いただきました岩下克氏，大内政義氏，日頃から様々なことを議論してきた，三武裕玄助教，高瀬裕氏，加藤史洋氏に深く感謝いたします。

電気通信大学在学の頃には，様々な視点から私の研究テーマについてご助言いただいた松野文俊教授（現京都大学教授），稲見昌彦教授（現慶應大学教授），横井浩史教授に深く感謝いたします。また，研究テーマについて，池田有冬氏，松永昇悟氏にはお世話になりました。両氏のおかげで力覚インタラクションのデモを作り，展示することができました。合わせて授業から国際学生対抗バーチャルリアリティコンテストでの作品展示まで，多くのことで支えあった同期の椎名美奈氏，ダニアルケオキ氏，時崎崇氏の皆様にも深く感謝いたします。

本論文の審査を通して長橋宏教授，石井秀明准教授，小野功准教授には貴重なご意見を頂きました。深く感謝いたします。

最後に，日頃から研究生活を共にした，東京工業大学の佐藤誠・長谷川晶一研究室の皆様にお礼申し上げます。

参考文献

- [1] Shoichi Hasegawa and Makoto Sato. Real-time rigid body simulation for haptic interactions based on contact volume of polygonal objects. *Computer Graphics Forum*, Vol. 23, No. 3, pp. 529–538, 2004.
- [2] S. Kim, J. Berkley, and M. Sato. A novel seven degree of freedom haptic device for engineering design. *VIRTUAL REALITY*, Vol. 6, No. 4, pp. 217–228, 2003.
- [3] 川崎晴久, 堀匠, 毛利哲也. 対向型多指ハプティックインターフェイス. *日本ロボット学会誌*, Vol. 23, No. 4, pp. 449–456, 2005.
- [4] Norio Nakamura and Yuiko Fukui. An innovative non-grounding haptic interface 'gyrocubesensuous' displaying illusion sensation of push, pull and lift. *SIGGRAPH '05 ACM SIGGRAPH 2005 Posters*, 2005.
- [5] 青木孝文, 三武裕玄, 長谷川晶一, 佐藤誠. ワイヤによる皮膚感覚刺激を用いた指先装着型接触感提示デバイス. *日本バーチャルリアリティ学会論文誌*, Vol. 14, No. 3, pp. 421–428, 2009.
- [6] 足立吉隆, ホウステファン, シャルマサンジェイ, 池本明夫, 熊野宜弘. Haptic cam システムの開発. *精密工学会誌*, Vol. 68, No. 4, pp. 591–596, 2002.
- [7] 国立情報学研究所. Hapticchem. <http://research.nii.ac.jp/~cheminfo/HapticChem/>.
- [8] MOOG Inc. Moog simodont dental trainer. <http://www.moog.com/markets/medical-dental-simulation/haptic-technology-in-the-moog-simodont-dental-trainer/>.
- [9] Novint Technologies Inc. Novint falcon. <http://www.novint.com>.

-
- [10] M. Moore and J. Wilhelms. Collision detection and response for computer animation. *Computer Graphics(SIGGRAPH 88)*, Vol. 22, pp. 289–298, 1988.
- [11] Y. J. Kim, M. A. Otaduy, M. C. Lin, and D. Manocha. Six-degree-of-freedom haptic display using incremental and localized computations. *Presence*, Vol. 12, , 2003.
- [12] E. Gilbert, D. Johnson, and S. Sathiy Keerthi. A fast procedure for computing the distance between complex objects in threedimensional space. *IEEE Journal of Robotics and Automation*, Vol. 4, No. 2, pp. 193–203, 1988.
- [13] G. van den Bergen. A fast and robust gjk implementation for collision detection of convex objects. *Journal of Graphics Tools*, Vol. 4, No. 2, pp. 7–25, 1999.
- [14] D. E. Muller and F. P. Preparata. Finding the intersection of two convex. *Theoretical Computer Science*, Vol. 7, No. 2, pp. 217–236, 1978.
- [15] David Baraff. Dynamic simulation of non-penetrating rigid bodies(ph. d thesis). *Computer Science Department, Cornell University*, pp. 52–56, 1992.
- [16] Erwin Coumans. Bullet physics library. <http://bulletphysics.org>.
- [17] Shoichi Hasegawa and Springhead Development Team. Springhead2. <http://springhead.info/wiki/>.
- [18] Ming C. Lin and John F. Canny. A fast algorithm for incremental distance calculation. In *IEEE International Conference on Robotics and Automation*, pp. 1008–1014, 1991.
- [19] S.Redon, A. Kheddar, and S. Coquillart. Fast continuous collision detection between rigid bodies. In *Computer Graphics Forum (Eurographics 2002 Proc.)*, Vol. 3, No. 21, 2002.
- [20] Khaled Mamou and Faouzi Ghorbel. A simple and efficient approach for 3d mesh approximate convex decomposition. *2009 16th IEEE International Conference on Image Processing (ICIP)*,, pp. 3501–3504, 2009.

- [21] C. Ericson. ゲームプログラミングのためのリアルタイム衝突判定. 株式会社ボーンデジタル, 2006.
- [22] B. Mirtich. Impulse-based dynamic simulation of rigid body system. *ph.D. dissertation, University of California, Berkeley*, 1996.
- [23] 足立吉隆. 力覚提示が可能なデジタルモックアップシステム. 日本設計工学会誌, Vol. 41, No. 3, pp. 122–131, 2006.
- [24] Sensable. Phantom. <http://www.sensable.com/index.htm>.
- [25] 横小路泰義, 村守宣彦, 吉川恒夫. 多指遭遇型ハプティックデバイスの設計. 日本バーチャルリアリティ学会第7回大会論文集, pp. 69–72, 2002.
- [26] S. Walairacht, K. Yamada, S. Hasegawa, Y. Koike, and M.Sato. Two-handed multi-fingers virtual object manipulation with haptic cue. *Transactions of IEICE-DII*, Vol. J84-D-II, No. 6, pp. 1140–1148, 2004.
- [27] 家室証, 南澤孝太, 舘すすむ. スケッチ入力によるモデリングのための非接地ペン型力覚ディスプレイ. 日本バーチャルリアリティ学会論文誌, Vol. 16, No. 3, pp. 459–468, 2011.
- [28] 南澤孝太, 家室証, 川上直樹, 舘すすむ. 指先装着型触力覚ディスプレイを用いた空中におけるVR物体の位置と大きさの提示. 日本バーチャルリアリティ学会論文誌, Vol. 13, No. 4, pp. 415–420, 2008.
- [29] CyberGlove Systems. Cybergrasp. <http://www.cyberglovesystems.com/products/cybergrasp/overview>.
- [30] J. Edward Colgate, Michael C. Stanley, and J. Michael. Issues in the haptic display of tool use. *IEEE/RSJ International Conference on Intelligent*, pp. 140–145, 1995.
- [31] Loonie Love and Wayne Book. Contact stability analysis of virtual walls. *Proc. Of Dynamic Systems and Control Division ASM*, pp. 689–694, 1995.

-
- [32] Y. Adachi, T. Kumano, and K. Ogino. Intermediate representation for stiff virtual objects. *Virtual Reality Annual International Symposium*, pp. 203–210, March 1995.
- [33] 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.
- [34] William R. Mark, Scott C. Randolph, Mark Finch, James M. Van Verth, and Russell M. TaylorII. 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.
- [35] Shoichi Hasegawa, Masahiro Ishii, Yasuharu Koike, and Makoto Sato. Inter-process communication for force display of dynamic virtual world. *Proc. of the ASME-Dynamic Systems and Control Division-1999*, Vol. 67, pp. 211–218, 1999.
- [36] Miguel A. Otaduy and Ming C. Lin. A modular haptic rendering algorithm for stable and transparent 6-dof manipulation. *IEEE Transactions on Robotics*, Vol. 22, No. 4, pp. 751–762, August 2006.
- [37] Katsuhito Akahane, Shoichi Hasegawa, Yasuharu Koike, and Makoto Sato. A proposal of a high definition haptic rendering for stability and fidelity. *ICAT2006*, pp. 162–167, November 2006.
- [38] Rene Weller and Gabriel Zachmann. Inner sphere trees for proximity and penetration queries. *Robotics Science and Systems Conference (RSS)*, 2009.
- [39] David Baraff. Fast contact force computation for nonpenetrating rigid bodies, 1994.
- [40] Diego Ruspini, Krasimir Kolarov, and Oussama Khatib. The haptic display of complex graphical environments. *SIGGRAPH 97 Proceedings*, 1997.

- [41] 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.
- [42] 菊植亮, 藤本英雄. 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*, Vol. 25, No. 2, pp. 142–151, 3 2007.
- [43] Peter J. Berkelman and Ralph L. Hollis. Lorentz magnetic levitation for haptic interaction: Device design, performance, and integration with physical simulations. *The International Journal of Robotics Research*, Vol. 19, No. 7, pp. 644–667, 2000.
- [44] William A. Mcneely, Kevin D. Puterbaugh, and James J. Troy. Six degree-of-freedom haptic rendering using voxel sampling. In *Proc. of ACM SIGGRAPH*, pp. 401–408, 1999.
- [45] 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.
- [46] Jonathan D. Cohen, Ming C. Lin, Dinesh Manocha, and Madhav Ponamgi. I-collide: an interactive and exact collision detection system for large-scale environments. *Proc. of the 1995 symposium on Interactive 3D graphics*, 1995.
- [47] Brian Mirtich. V-clip: fast and robust polyhedral collision detection. *ACM Transactions on Graphics*, Vol. 17, pp. 177–208, 1998.
- [48] 横山哲也, 飯田桂弘. 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*, Vol. 4, pp. 51–56, 2002.
- [49] L. Glondu, M. Marchal, and G. Dumont. A new coupling scheme for haptic rendering of rigid bodies interactions based on a haptic sub-world using a contact graph. *EuroHaptics'10 Proceedings of the 2010 international conference on Haptics*, pp. 51–56, 2010.

- [50] 永井岳大, 星野崇宏, 内川恵二. 恒常法により推定された閾値間の統計的有意差検定法. *日本視覚学会誌 Vision*, Vol. 18, pp. 113–124, 2006.
- [51] Allison M. Okamura, Marak R. Cutkosky, and Jack Tigh Dennerlein. Reality-based models for vibration feedback in virtual environments. *IEEE/ASME Transactions on Mechatronics*, Vol. 6, Issue: 3, pp. 245–252, 2001.
- [52] Yuto Ikeda and Shoichi Hasegawa. Short paper: Characteristics of perception of stiffness by varied tapping velocity and penetration in using event-based haptic. *Joint Virtual Reality Conference EGVE-ICAT-EURO VR*, pp. 113–116, 2009.
- [53] 赤羽克人, 高見豪, 佐藤誠. フレームの可動機構を持つワイヤ駆動多指型ハプティックインタフェースの開発. *日本バーチャルリアリティ学会論文誌*, Vol. 16, No. 3, pp. 441–448, 2011.
- [54] 鎌田勝寛, 竹花幸伸, 藤田欣也. 共同仮想物体操作のための遠隔作業者間の手の拘束の力覚提示. *日本バーチャル・リアリティ学会大会第16回大会論文集, セッション12C*, , 2011.
- [55] Rene Weller and Gabriel Zachmann. User performance in complex bi-manual haptic manipulation with 3 dofs vs. 6 dofs. *Haptics Symposium 2012*, pp. 315–322, 2012.
- [56] 本間達, 若松秀俊. 臓器および操作器具の数学モデルによる医学教育用力覚表示システムの開発. *日本バーチャルリアリティ学会第4回力触覚の提示と計算研究会報告書*, Vol. 1, No. HDC04, pp. 11–14, 2010.
- [57] Fumihiko Kato, Yusuke Hanaoka, Tu Nguyen Ngoc, Danial Keoki, Hironori Mitake, Takafumi Aoki, and Shoichi Hasegawa. Interactive cooking simulator. *ACM SIGGRAPH 2009 Emerging Technologies*, No. 14, 2009.
- [58] Min Tang, Dinesh Manocha, Miguel A. Otaduy, and Ruofeng Tong. Continuous penalty forces. *Proc. of ACM SIGGRAPH 2012*, 2012.
- [59] Hironori Mitake, Shoichi Hasegawa, and Makoto Sato. Reactive virtual creatures for dexterous physical interactions. *Motion in Games 2011*, 2011.

- [60] 黒田嘉宏, 仲谷正史, 長谷川晶一, 藤田欣也. 物理的刺激に基づく疑似力覚の提示と計算に関する研究動向. 日本バーチャルリアリティ学会論文誌, Vol. 16, No. 3, pp. 379–390, 2011.

関連論文の印刷公表の方法及び時期

学術雑誌論文（査読付き）

1. 全著者名：須佐 育弥, 大内 政義, 岩下 克, 佐藤 誠, 長谷川 晶一
論文題目：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提示の実現
印刷公表の方法および時期：日本バーチャルリアリティ学会論文誌 第14巻第4号, pp463-471, 2009
(第4章の内容に関連する)
2. 全著者名：須佐 育弥, 長谷川 晶一
論文題目：6自由度力覚レンダリングのための中間表現
印刷公表の方法および時期：日本バーチャルリアリティ学会論文誌 第17巻第1号, pp.45-57, 2012
(第3章の内容に関連する)

国際会議

口頭発表・技術展示（査読付き）

1. 全著者名：Ikumi Susa, Makoto Sato, Shoichi Hasegawa
論文題目：Multi-Rate Multi-Range Dynamic Simulation for Haptic Interaction
印刷公表の方法および時期：Proc. of the World Haptics Conference2011, 2011
(第4章の内容に関連する)

技術展示（アブストラクト査読付き）

1. 全著者名：Ikumi Susa, Yuto Ikeda, Takashi Tokizaki, Hironori Mitake, Makoto Sato, Shoichi Hasegawa
論文題目：Perception-Based High-Definition Haptic Rendering
印刷公表の方法および時期：SIGGRAPH 2008 new tech demos, Article No.32, 2008
(第4章の内容に関連する)

技術展示（査読なし）

1. 全著者名：Ikumi Susa, Yuto Ikeda, Shoichi Hasegawa
論文題目：Perception-based High Definition Haptic Rendering
印刷公表の方法および時期：JVRC09-Joint Virtual Reality Conference of EGVE-ICAT-EuroVR Demonstration Program, pp.49-50, 2009.
（第4章の内容に関連する）
2. 全著者名：Ikumi Susa, Yuto Ikeda, Takashi Tokizaki, Hironori Mitake, Shoichi Hasegawa
論文題目：Haptic Rendering for Rigid Body Simulator based on Analytical Methods
印刷公表の方法および時期：EuroHaptics 2008 hands on demo, 2008.
（第4章の内容に関連する）

国内学会大会・研究会

口頭発表（査読付き）

1. 全著者名：須佐 育弥, 池田 有冬, 三武 裕玄, 佐藤 誠, 長谷川 晶一
論文題目：Perception-Based High-Definition Haptic Rendering
印刷公表の方法および時期：インタラクティブ東京 2008, 日本科学未来館, 2008
9.
（第4章の内容に関連する）

口頭発表（査読なし）

1. 全著者名：須佐 育弥, 長谷川 晶一
論文題目：マルチレートな剛体力覚提示システムのための6自由度力覚レンダリング手法の提案
印刷公表の方法および時期：日本バーチャルリアリティ学会第15回大会論文集, セッション 3D2-4, 2010.
（第3章の内容に関連する）

2. 全著者名：須佐 育弥, 長谷川 晶一
論文題目：中間表現を用いた 6 自由度力覚レンダリング手法の提案
印刷公表の方法および時期：力触覚の提示と計算 第 4 回研究会, 2010.
(第 3 章の内容に関連する)
3. 全著者名：須佐 育弥, 長谷川 晶一
論文題目：大規模バーチャル世界のための高品質力覚提示
印刷公表の方法および時期：日本バーチャルリアリティ学会第 14 回大会論文集, pp.2A2-1, 2009.
(第 4 章の内容に関連する)
4. 全著者名：須佐 育弥, 長谷川 晶一
論文題目：高解像度力覚レンダリングのための高速物理シミュレーション 印刷
公表の方法および時期：力触覚の提示と計算研究委員会 第 1 回研究会, pp.15-
18, 2009.
(第 4 章の内容に関連する)
5. 全著者名：須佐 育弥, 長谷川 晶一
論文題目：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
印刷公表の方法および時期：ヒューマンインタフェース学会研究報告集, Vol.10,
No.2, pp.57-62, 2008.
(第 4 章の内容に関連する)

参考論文の印刷公表の方法及び時期

参考論文

1. 全著者名：須佐 育弥, 高橋 悠太郎, 池田 有冬, 長谷川 晶一, 佐藤 誠
論文題目：振動を付加させた力覚提示における硬さの知覚
印刷公表の方法および時期：力触覚の提示と計算 第7回研究会, 2011.
2. 全著者名：長谷川 晶一, 三武 裕玄, 青木 孝文, 須佐 育弥, 松山 隼輔, 土田 隆太郎
論文題目：多様なインタフェース・インタラクションに対応したキャラクタ動作の生成
印刷公表の方法および時期：エンタテインメントコンピューティング 2010, セッション 2, 2010.

受賞

1. 情報処理学会エンタテインメントコンピューティング優秀論文賞 (2007)

著者略歴

須佐 育弥 (すさ いくみ)

1985年 大阪府に生まれる

2004年4月 電気通信大学 電気通信学部 知能機械工学科 入学

2008年3月 電気通信大学 電気通信学部 知能機械工学科 卒業

2008年4月 電気通信大学大学院 電気通信学研究科
知能機械工学専攻 博士前期課程 入学

2010年3月 電気通信大学大学院 電気通信学研究科
知能機械工学専攻 博士前期課程 修了

2010年4月 電気通信大学大学院 情報理工学研究科
知能機械工学専攻 博士後期課程 入学

2010年9月 電気通信大学大学院 情報理工学研究科
知能機械工学専攻 博士後期課程 退学

2010年10月 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 転入学

2012年9月 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 修了予定

2011年より日本学術振興会特別研究員 (DC2), バーチャルリアリティ, 力覚インタラクションの研究に従事。2007年情報処理学会インタテイメントコンピューティング優秀論文賞。