

ハプティックインタラクションのための物理ベースモデリング

Physically Based Modeling for Haptic Rendering

長谷川晶一¹⁾, 田崎勇一²⁾, 佐藤誠¹⁾

Shoichi HASEGAWA, Yuuichi TAZAKI, Makoto SATO

1) 東京工業大学精密工学研究所

(〒226-8503 横浜市緑区長津田町 4259, {hase@hi., msato@}pi.titech.ac.jp)

2) 東京工業大学情報環境学専攻

(〒152-8552 東京都目黒区大岡山 2-12-1, tazaki@cyb.mei.titech.ac.jp)

Abstract : We propose a simulator which can simulate articulated bodies and multi bodies in haptic rates. The simulator can realize haptic interaction with articulated bodies like animal bodies or robots. We employ penalty method for normal and friction forces and Featherstone's method for joint forces. Both methods are very fast ($O(n)$). We perform some experiments and they show that our simulator runs in haptic rates and realizes haptic interaction with multi bodies.

Key Words: *Haptic, Physically based modeling, rigid body, articulated body*

1. はじめに

物理法則を基に仮想物体の運動をシミュレーションし、力覚インタフェースを介してインタラクションすることで、リアリティの高い仮想物体との直感的なインタラクションが実現できる。

我々は、剛体の接触体積に基づくペナルティ法を用いた剛体運動のシミュレーション手法を提案している [1]。本稿ではこれを発展させ、蝶番や関節を持つ物体を含む運動シミュレーション手法を提案する。これは、例えば地面の上を歩く仮想の動物やロボットとの力覚インタラクションを実現する。

2. 剛体運動のシミュレーションと従来の研究

2.1 接触力の計算

物体の運動は、物体に働く力が求められれば、運動方程式の数値解として求まる。重力、バネによる力、動摩擦力など、場の力や物体の位置・速度によって決まる力は簡単な計算で求まる。しかし、関節に働く力、抗力、静止摩擦力など複数物体の位置関係を拘束する力（拘束力）の計算はそれほど簡単ではない。

接触力の計算 接触力の計算法として、接触力を直接求める解析的な手法と、バネダンパモデルを用いてシミュレーションの繰り返しの中で間接的に求めるペナルティ法が提案されている。

Baraff ら [2] [3] は、運動方程式と拘束条件から接触力を解く手法を提案している。Baraff らの手法は接触の数 n に対して $O(n^3)$ の計算時間を要する。

Mirtich と Canny [4] は抗力を衝突時に 2 物体に働く撃力によって表現する手法を提案している。この手法は衝突が発生した時刻を求め、2 物体の衝突を順に処理する。この手法は、1 衝突の処理にかかる計算時間は短いですが、短期間に多数の接触が起きる場合多くのステップを処理する必要がある。そのため、一定期間のシミュレーションに多大な計算時間を要することがある。

以上のように、解析的な手法はシミュレーションに必要な計算時間が大きく変動するため、力覚提示に必要な実時間性を維持できない。

Moore と Wilhelms [5], McKenna と Zeltzer [6], Keller ら [7] はペナルティ法を用いて接触力を求めている。ペナルティ法とは、拘束を侵した物体に拘束を侵した量 (= ペナルティ) に応じた力を与えることで拘束違反を解消する方法である。接触力の計算にペナルティ法を用いると、接触の数 n に対して計算時間が $O(n)$ となり、高速処理が可能である。しかし従来の研究では、物体が面と面で接した場合に抗力が不安定になる、摩擦力の計算が正確でない、といった問題があった。そこで我々は、接触体積に基づく抗力計算法を提案した [1]。

関節の拘束力の計算 動物の体やロボットのような関節を持つ物体は、複数の剛体（リンク）が関節で結合されると考えることができる。

このような物体では、シミュレーション中にリンクや関節の数、接続状態が変化しないため、拘束条件式も変化しない。また抗力や摩擦力のように、拘束条件に不等式が入ることはなく、拘束条件は等式で表される。そのため、関節

の拘束力は、抗力や静止摩擦力に比べて高速に計算することができる。

例えば、関節が輪を持たない場合について、Featherstone [8] が高速 ($O(n)$) な計算法を提案している。

3. 提案

力覚提示に必要な高速更新を実現するため、次のような構成のシミュレータを提案する。

- 抗力、摩擦力の計算は接触体積に基づくペナルティ法 [1] を用いる。
- 関節に働く拘束力の計算には、Featherstone の方法 [8] を用いる。

関節に働く拘束力もペナルティ法を用いて求めることもできるが、ペナルティ法は、拘束を侵した量に応じた力を与える方法なので、拘束が正確に満たされない。抗力や摩擦力の場合も、これが不自然な動きの原因になることがあるが、関節の場合、関節が複数直列につながっている場合などにこの誤差が重畳されるため、非常に不自然な動きになることがある。構造は制限されるが、関節に働く拘束力は Featherstone の手法で高速に直接計算できるので、2 つの手法を組み合わせて、高速で自然な動きを実現する。

4. 実装

提案手法に基づいてシミュレータを構築した。本節では提案シミュレータの動作を説明する。

4.1 シミュレーションの流れ

提案シミュレータは、次の計算を行う。

1. 重力など場の力の計算
2. 接触判定
3. 接触体積と抗力、摩擦力の計算
4. Featherstone の方法 [8] による関節に働く拘束力の計算
5. 加速度、速度を積分し、速度、位置を求める。

4.2 接触力の計算

物体に働く接触力は、接触の法線に平行な抗力と、垂直な摩擦力に分けて考える。

抗力 接触面には、接触面上の任意の点が互いに侵入しないという拘束を満たすような力、すなわち抗力が働く。そこで、接触領域全体にバネダンパモデルが分布すると考え、バネダンパモデルが発生する力の総和が物体に働いたと考える。これにより接触面に働く力とトルクを計算する。

摩擦力 接触面上の微小面積に働く動摩擦力や最大静止摩擦力は、その微小面積に働く抗力に比例する。接触面全体に働く摩擦力はこれらの総和となる。

そこで接触領域全体にクーロンの摩擦モデルが分布し、摩擦力を発生すると考える (図 1)。これにより、摩擦力だけでなく、摩擦トルクも計算できる。

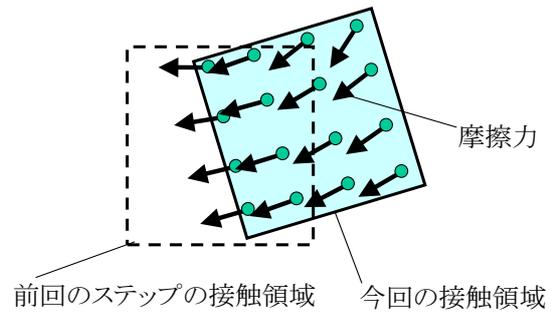


図 1: 摩擦力のモデル

接触体積の計算 提案シミュレータでは、物体形状を凸多面体の集合で表し、凸多面体単位で接触判定を行い、接触体積を求める。

提案シミュレータでは、データ構造が単純で計算が高速な GJK アルゴリズム [9] を用いた。GJK アルゴリズム [9] は、凸形状対が与えられたときに、最近傍点对もしくは接触体積上の 1 点を求める。

Muller と Preparata [10] は、凸多面体の共通部分の内部の点が 1 点与えられたときに、共通部分全体 (を構成する 3 角形) を求めるアルゴリズムを提案している。

提案シミュレータでは Muller のアルゴリズムを用いて共通部分を構成する 3 角形を求め、3 角形ごとに抗力や摩擦力を計算する。

4.3 関節の拘束力の計算

関節に働く拘束力の計算には Featherstone の方法 [8] を用いた。ここでは簡単にこの方法を説明する。

木構造のリンクでは、あるリンクは 1 つのより根に近い側のリンクといくつかのより葉に近い側のリンクと関節で接続している。

あるリンク i が根側のリンクから受ける力とトルク (= 拘束力) を \hat{f}_i^I 、リンクの加速度と角加速度を \hat{a}_i とすると、

$$\hat{f}_i^I = \hat{I}_i^A \hat{a}_i + \hat{Z}_i^A \quad (1)$$

のように、リンク i とより葉側のリンク全体が一つの剛体であるかのような式が書け、 \hat{I}_i 、 \hat{Z}_i は関節加速度によらない (図 2 参照。証明は Mirtich [4] p.106 を参照)。そこで、

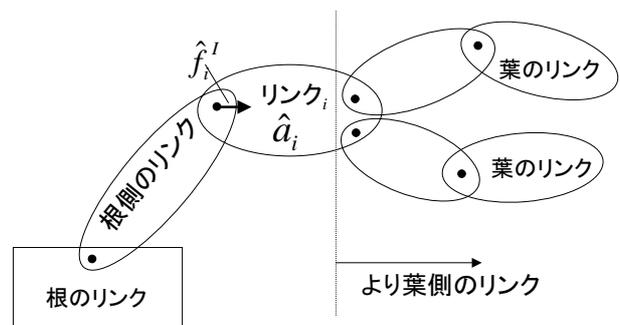


図 2: 木構造のリンク

次のように、拘束力を求める。

1. 木構造の根から葉に向かって、各リンクの速度と角速度を求める。
2. 木構造の葉から根に向かって、式 (1) の \hat{f}_i^A, \hat{Z}_i^A を求める。また、このとき拘束力 \hat{f}_i^f も求める。
3. 木構造の根から葉に向かって、各関節の加速度と、リンクの加速度・角加速度を求める。

このように Featherstone の方法では、拘束力と同時に各リンクの加速度・角加速度を求めることができる。また、2. のときに、各リンクにペナルティ法で求めた接触力を加えることで、接触力を考慮に入れた各リンクの加速度を求めることができる。

5. 評価

提案シミュレータを評価するため、いくつかの実験を行った。

5.1 抗力の計算時間の評価

提案手法が、Baraff[2] [3] などが提案する拘束を解く方式に比べて、高速で力覚提示に向いていることを確認するために、計算時間を比較した。拘束を解く方式のシミュレータとしては、Open Dynamics Engine [11] を用いた。

方法 両方のシミュレータで図 3 のように積み上げられたブロックをシミュレートし、ブロックの数と 1 ステップの計算時間の平均の関係を調べた。1 ステップのシミュレータ内の時間は 5ms、重力加速度を $9.8m/s^2$ 、ブロックの大きさを $1m \times 1m \times 2m$ 、質量を 1kg とした。シミュレーションは、Intel Pentium 4 1.6GHzCPU を 1 個搭載した PC で行った。

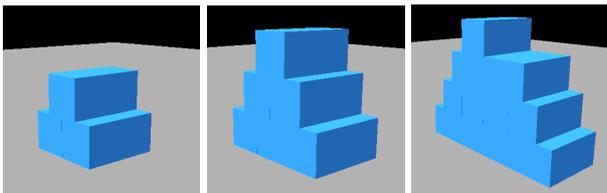


図 3: 評価のために用いたシーン：3 個，6 個，13 個のブロック

結果 図 4 に、ブロックの数と 1 ステップの計算時間の関係を示す。拘束を解析する Open Dynamics Engine では、ブロックの数が増えるにつれて、急激に計算時間が増加するが、提案シミュレータでは、ほぼ線形に計算時間が増加する。

以上から、特にブロック数がある程度多くなったときには、ペナルティ法の方が 1 ステップのシミュレーションに必要な計算時間が短く、接触点数の増加に対する計算時間の増加が緩やかなので、力覚提示に向いていることが分かる。

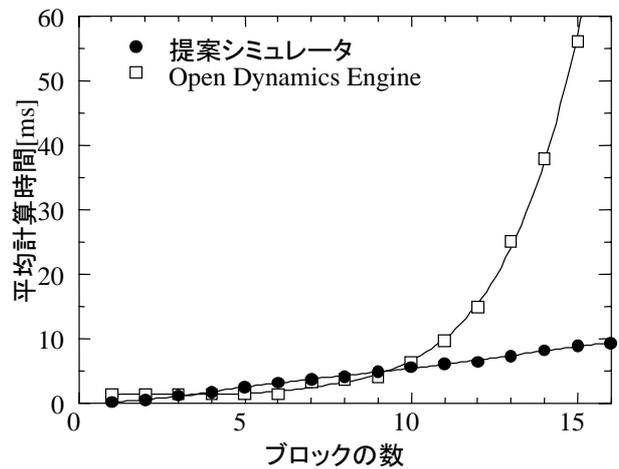


図 4: 計算時間の比較

5.2 関節を持つ物体のシミュレーション

関節を持つ物体を高速にシミュレーションできることを確かめるため、剛体と 1 自由度の関節からなる鎖のシミュレーションを行った。図 5 にシミュレーションされた鎖の画像を示す。このシミュレーションに Pentium III 750MHz

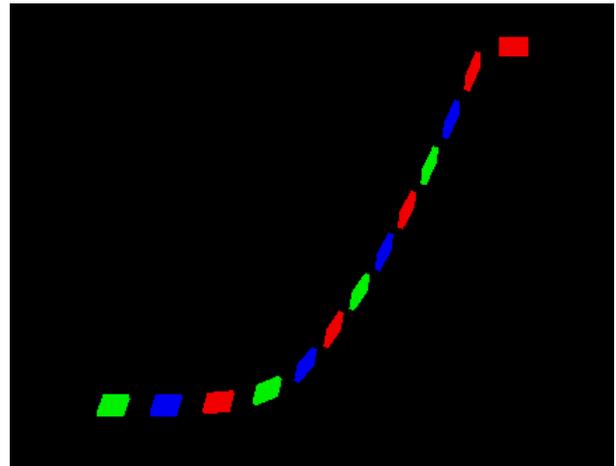


図 5: 鎖のシミュレーション

の CPU を持つ PC を用いたところ、1kHz の更新周期を維持することができた。

5.3 両手直接操作環境

提案シミュレータと力覚インタフェースを用いて、直接操作環境を構築できることを確かめた。実験には力覚インタフェース SPIDAR [12] を 2 つ用いて 6 個の積み木が置かれた仮想世界を提示し、直接操作が行えることを確かめた。またこのとき各処理の計算時間を計測した。この結果、積み木を動かすなどの直接操作を直感的に行えることが確認できた。また、接触判定と接触力の総和を求める計算に多くの計算時間が必要なことがわかった。図 6 に積み木を崩したあと、再び積み上げる様子と計算時間の推移を示す。このシミュレーションは Pentium4 2.8GHz CPU を 1 つ搭

載した PC で行った .

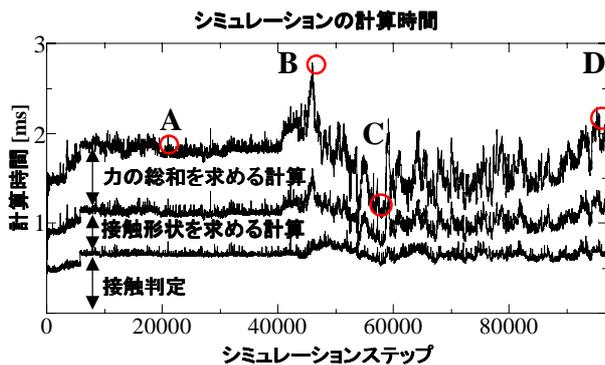
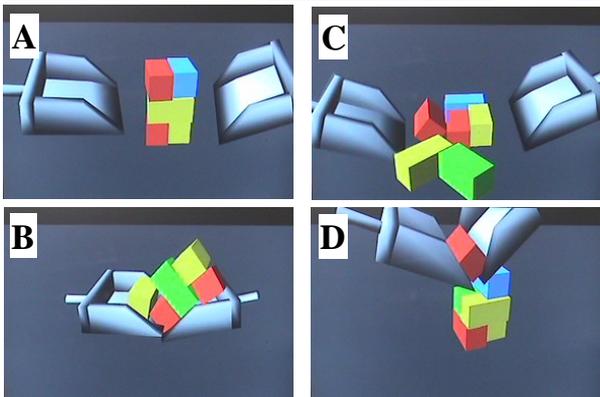


図 6: 力覚インタフェースによる直接操作と計算時間

6. おわりに

動物やロボットのような関節を持つ物体と力覚インタフェースを介してインタラクションする方法を提案した . 今後は , 提案した方式を用いて , 仮想の動物や歩行ロボットに力覚インタフェースを介して触れることができる環境を作りたいと考えている .

参考文献

- [1] 藤井伸旭, 長谷川晶一, 橋本直己, 小池康晴, 佐藤誠: “ペナルティー法を用いた剛体運動シミュレータの開発”, 日本バーチャルリアリティ学会第 7 回大会論文集
- [2] D. Baraff: “Analytical methods for dynamic simulation of non-penetrating rigid bodies”, Computer Graphics Proceedings(SIGGRAPH 89) Vol. 23, pp. 223-232, 1989.
- [3] D. Baraff: “Fast contact force computation for non-penetrating rigid bodies”, Computer Graphics Proceedings(SIGGRAPH94), pp. 23-34, 1994.
- [4] B. Mirtich: “Impulse-based Dynamic Simulation of Rigid Body Systems”, UC Berkeley PhD Thesis, Fall 1996.
- [5] M. Moore, J. Wilhelms: “Collision Detection and Response for Computer Animation”, Computer Graphics (SIGGRAPH 88), Vol. 22, pp. 289-298, August 1988.
- [6] M. McKenna and D. Zeltzer: “Dynamic simulation of autonomous legged locomotion”, Computer Graphics (SIGGRAPH 90), Vol. 24, pp. 29-38, August 1990.
- [7] H. Keller, H. Stolz, A. Ziegler: “Virtual Mechanics: Simulation and Animation of Rigid Body Systems”, <http://citeseer.nj.nec.com/keller94virtual.html>, 38 pages, 1993.
- [8] R. Featherstone: The Calculation of Robot Dynamics using Articulated-Body Inertias, Int. J. Robotics Research, vol. 2, no. 1, pp. 13-30, 1983.
- [9] E. G. Gilbert, D. W. 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), 1988, pp. 193-203.
- [10] D. E. Muller, F.P.Preparata, “Finding the intersection of two convex polyhedra,” Theoretical Computer Science, 7(2), 1978, pp. 217-236.
- [11] Russell Smith: Open Dynamics Engine, <http://opende.sourceforge.net/>
- [12] Masahiro Ishii and Makoto Sato: A 3D Spatial Interface Device Using Tensed Strings, Presence Vol. 3, No. 1, Winter 1994, pp.81-86