

変形物体のための高速な接触検知手法

坂本 英男[†] 長坂 学[†] 緒方 正人[†]

本論文では、物体間の接触判定を高速に行う手法について述べる。弾性体間やその自己接触を含む変形計算を実時間で行うには、変形する接触対象を高速に検出する必要がある。提案手法では、有限要素モデルで規定された物体を、2進木を基に構造化することにより、高速に接触部を検知する。その結果、実用的な規模の計算機で、4物体各2,500節点に対して、実時間(30Hz)での弾性体間の変形計算が可能となった。

A Fast Collision Detection for Deformable Objects

HIDEO SAKAMOTO,[†] MANABU NAGASAKA[†] and MASATO OGATA[†]

In this paper, we propose a fast collision detection method for deformable objects. To simulate deformation of interaction between these objects and self collision objects, it is necessary to detect contacting nodes in realtime. Structuring finite element objects with binary tree, we achieved fast detection of contacting nodes. As a consequence of the proposed method, we performed realtime deformation with a computer of reasonable size for 4 elastic objects each has 2,500 nodes.

1. はじめに

筆者らは、2005年より専門医の手術手技訓練が可能なる手術シミュレータを開発している。この開発の主要課題として多臓器の変形計算を実時間で行う必要がある。この計算では、外力を原因としてその結果である変位を計算するために、線形弾性体モデルに基づいて有限要素法により計算している³⁾。これにより、数学的には、外力が判明すればその結果として、変位を得ることが可能となる。しかし、臓器を通して、他の別の臓器に力が伝わる複合臓器や自臓器が自己の他の部分に接触して力が伝わる自己接触の場合などにおいては、この外力が直接的に計算できない。従って、専門医の手術手技訓練を目的とした手術シミュレータの開発にあたっては、これらの相互作用をある程度の正確さで実時間シミュレーションすることが大きな課題である。幸いなことに、このような場合の取り扱いには250年前の天才達のアイデアを端緒として理論的には把握されている。例えば、相互に接触する点が分れば、この外力はラグランジュ法やペナルティ法により間接的に求められるので、前述のように変位が計算できる。従って、複合臓器間の接触や自己接触による変形を実現するには、物理的な計算の前に、相互に接触する点を検出することが必要となる。このときの困難さは理論的なものではなく実時間処理に起因する計

算工学的な困難さである。即ち、時々刻々と変化する臓器を構成する膨大な量の節点間での接触点の検出を33ms(あるいは16ms)内に行う困難さである。

例えば、計算の複雑さの例として、ポリゴン数(表面を構成する有限要素数) N のモデル同士の接触判定の計算量を考えると、単純な方法では $O(N^2)$ 回の接触計算を行わなければならない、 N が大きい場合、非常に計算コストの高い処理となる。我々が対象としている手術シミュレータでは数万節点を持つモデルが必要であり、この接触節点群の検出を実時間で行うには工夫が必要である。

我々は、この大規模なモデル同士の接触判定を高速に処理するために、有限要素モデルで規定された各臓器モデルを木構造を用いて構造化することにより、判定処理の対象ポリゴン数 M を $M \ll N$ とすることができた。これにより、処理時間を大幅に削減した。現在、4物体(臓器)各2,500節点に対して、実時間(30Hz)での力学計算が可能となった。この構造化データは、生体モデル記述言語としてテキスト形式で記述され、力学計算と映像発生に共用できるので、手術シミュレータのみならず、他の汎用的な実時間力学シミュレーション、例えば衝突シミュレーションなどにも応用可能である。

本論文の構成は2章において従来研究を述べ、3章に提案手法、4章にその評価、最後に5章に結論を述べる。

[†]三菱プレジジョン株式会社
Mitsubishi Precision Co., Ltd.

2. 従来研究

接触判定手法の研究は、コンピュータグラフィックス (CG) の重要な技術課題であった隠蔽処理と同種の探索問題であることや、CG を応用した CAD (Computer Aided Design) で、物体間の干渉を高速に判定する必要から、CG の初期の段階から既に行われていた。例えば、最近の Bounding Volume Hierarchy (BVH) 法²⁾ は、既に、隠蔽処理として、1980 年代から 90 年代の戦闘機用フライトシミュレータのハイエンド CG で用いられている^{5)~7)}。従って、研究としては長い歴史を持つが、近年、力学的な実時間シミュレーション (ゲームも含む) 要求の高まりから、新たな視点での発展を見せている。

先ず、そのような接触判定の高速化手法^{1),8)} は、剛体モデルにおいては確立され、弾性体についても適用されつつある。主な手法としては、空間ハッシュ法⁹⁾ と BVH 法がある。

空間ハッシュ法は、シミュレーション空間を分割して、各分割空間にあるポリゴンをハッシュテーブルに登録する。接触判定では、このハッシュテーブルを用いて、各々の分割空間内に同時にあるポリゴンを高速に探索できる。対象ハッシュテーブルにあるポリゴン同士のみで、接触判定を行えばよいので、判定回数は削減される。この手法の問題は、分割する空間の大きさの決定で、分割する空間の大きさが粗すぎると、分割後の同一空間に全てのポリゴンが入ってしまい、接触計算回数は減らない。逆に細かすぎると、ポリゴン 1 つを占める空間が幾つも出来てしまい、メモリの無駄な消費が発生する。剛体の場合、ポリゴンの大きさは一定のため、シミュレーション前に適切な空間の大きさを決定することが出来る。代表的な空間ハッシュ法⁴⁾ においては、分割する空間サイズの最適化を行い、処理の高速化を行っている。しかし、モデルのポリゴン形状の大きさが一様でない場合、最適な空間分割が難しいため、任意物体に対する対応が困難である。

一方、BVH 法は、対象物体を構成するポリゴンを凸包するプリミティブな形状 (Bounding Volume: BV) で、近似し、探索回数を削減する手法である。さらに、いくつかの近傍の BV を凸包するより大きな BV を生成していき、最終的には 1 つの BV となるように階層構造を作ることにより探索の効率化が図れる。この階層構造 (木構造) を辿ることで、近似的にポリゴン同士の接触を容易に見つけることが出来る。剛体の場合は、シミュレーション前のオフライン処理として計算できるが、弾性体の場合、形状が変形するため、BV 情報をリアルタイムで更新する必要がある。弾性体に適用するために、リアルタイム更新する手法も提案されている。例えば、論文²⁾ では、BVH 法を用いており、物体の変形形状に応じて、BV を更新してい

```
-----  
// サンプルモデルデータ  
// 有限要素で規定された腎臓オブジェクトデータと  
// 同血管オブジェクトデータの合成を示す。  
-----  
BRANCH( //2 臓器の複合を示すノードタグ  
//-----  
// 以下の木に繋がる全ノードに対して ME で指定した移動、回転、拡大を規定  
//  
ME trans 0,0,0 //平行移動を示す (x, y, z の順)  
rot 1,0,0,0 //回転量を示す (x, y, z, u の順):クォータニオン表記  
scale 1,1,1 //拡大率 (x, y, z の順)  
}  
//-----  
// 腎臓の有限要素モデル設定  
TREE(  
ME trans 2,0,0  
rot 1,0,0,0  
scale 1,1,2  
PE 1 //並列化時のプロセッサ ID  
NE kidney.seg //オブジェクトデータファイル名(腎臓)  
}  
//-----  
// 大動脈の有限要素モデルの設定  
TREE(  
ME trans 0,-2.5,-1.4  
rot 1,0,0,0  
scale 1,1,2  
PE 2  
NE aorta.seg //オブジェクトデータファイル名(大動脈)  
})
```

図 1 臓器の複合を生体モデル記述言語により規定した生体モデルデータサンプル

る。しかし、物体の切断等、トポロジーの変換には対応できないといった欠点がある。本論文では、BVH 法を基本として、物体の切断等、トポロジーの変化にも対応する BVH 法を提案する。

3. 提案手法

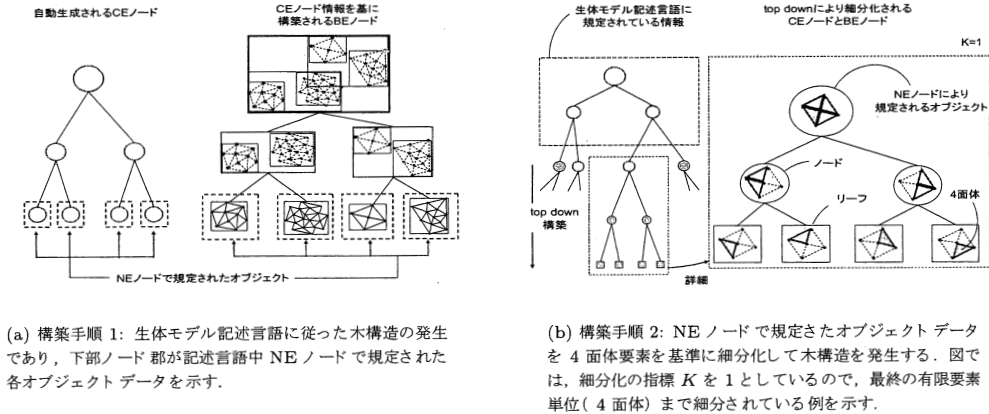
模擬対象モデルは、複数の構成臓器の部位 (部品) を組み合わせて構築するため、構造を規定するメタオブジェクト部と実体を規定したオブジェクトデータ部 (実有限要素データ) からなる。メタデータの構成要素は (1) バウンディングボリューム要素、(2) モディファイド要素、(3) コネクション要素、(4) ネームド要素、(5) プロセッサ要素の 5 種類からなる。表 1 に、メタデータの一覧とその機能をそれぞれ示す。また、図 1 に生体モデル記述言語で記述したモデリング例を示し、その出力映像を図 4(a) に示す。

3.1 2 進木を用いた構造化

2 進木を用いたメタオブジェクトによりデータの構造化を行う。このとき、2 進木の構築は次の 2 段階で行われる。言語で規定された物体間の接続は、構築手順 1 として、そのままメタオブジェクトの CE ノードに翻訳し、言語中の NE 文で規定されたオブジェクトデータをロードする。次に構築手順 2 として、ロードされたオブジェクトデータを構成する各 4 面体 (有限要素) を基に自動的に BE ノードおよび CE ノードを発生させ、2 進木 (メタオブジェクト) を作成する。以下、2 進木の終端要素をリーフ (Leaf)、それ以外をノード (Node) として、手順詳しく述べる。

表 1 記述言語の構文と対応するメタデータの種類および機能

メタデータ種類	略記	機能	備考
バウンディングエレメント	BE	下位の要素を凸包する 6 面体：ポリゴンを凸包する 6 面体	
モディファイドエレメント	ME	以下の木に対して回転，拡大，平行移動量を与える	初期化に使用
コネクションエレメント	CE	左右の木の連結（あるいは分岐）を示す	
ネームドエレメント	NE	オブジェクトデータ名（実体 FEM データ名）	
プロセッサエレメント	PE	オブジェクトを割り当てるプロセッサ ID	並列化時に使用



(a) 構築手順 1: 生体モデル記述言語に従った木構造の発生であり，下部ノード群が記述言語中 NE ノードで規定された各オブジェクトデータを示す。

(b) 構築手順 2: NE ノードで規定したオブジェクトデータを 4 面体要素を基準に細分化して木構造を発生する。図では，細分化の指標 K を 1 としているので，最終の有限要素単位（4 面体）まで細分されている例を示す。

図 2 木構造（メタオブジェクト）の 2 段階構築：木構造の発生は，生体モデル記述言語により規定された木構造の発生と，各オブジェクトに対応する木構造の自動発生の 2 段階からなる。

構築手順 1: 言語記述に従ったメタオブジェクトの発生

これは，生体モデル記述言語で規定された変形物体間の関係を表すメタオブジェクトを発生するステップである。まず，生体モデル記述言語に規定されている通りに，CE ノードを発生する。次に，NE ノードに記されている情報を基に変形物体のオブジェクトデータを作成する。このとき，NE ノードより上位に規定されている全 ME ノードの影響を受ける。最後に，CE ノードに従い，BE ノードをボトムアップにより発生させる。図 2(a) に模式図を示す。

構築手順 2: 自動分割によるメタオブジェクトの発生

次に，オブジェクトデータを細分割し，トップダウンにより木構造を構築する。図 2(b) に模式図を示す。細分割を行う基準は，木構造要素のメタデータである BE 内に含まれる有限要素単位である 4 面体数である。即ち，BE 内に含まれる 4 面体数が K 以上であった場合，BE の長軸方向 (x, y, z のいずれかに沿う) に対して，再度分割する。この処理を再帰的に行い，BE に含まれる 4 面体数が分割基準数 K 以下になるまで行う。分割基準となる 4 面体数 K は，任意だが，適切な値はアプリケーションによって適宜変更する必要がある。

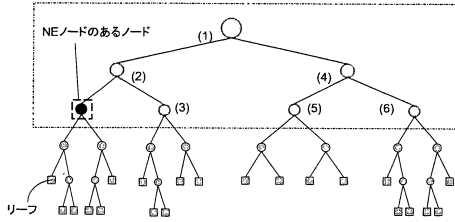
この木構造の構築は，物体の初期位置を基準にして，初期化時のみ行い，シミュレーション時は，木構造の再構築はせず，更新のみ行う。

3.2 接触部の検出：トラバース処理

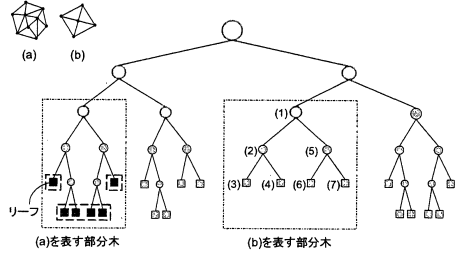
接触部の検出は物体を構成するポリゴンの空間的大きさの情報を持つメタデータ中の BE ノードを用いて行う。階層化したメタデータを用いて交差判定を行うことで，木探索の早い段階（塊りの大きな段階）で交差判定が可能となり，以後の不必要な交差判定を大きく削減できる。

この処理の前提として，2 進木の最下位（リーフ）にある BE ノード内のポリゴン間での自己接触は，お互いが十分近いため，変形の影響が少なく起こらないとしている。この前提は，必要に応じて最下位の BE ノードを 4 面体単位（1 有限要素）まで小さくすることが可能であるので妥当な仮定と考える。従って，同一 BE ノード内にあるポリゴン同士の接触判定は考える必要がない。

接触部の探索も 2 段階で行う。段階 1 はオブジェクトデータ単位であり，この処理を図 3(a) に具体的に示す。まず，木の先頭から物体の BE ノードを NE ノード検出まで辿り（NE ノード以下の BE ノードは辿らない），物体間同士の接触を検知する。段階 2 として，



(a) トラバース処理手順 1 (物体間の接触検知): 枠内は用いる要素を, (数字) は辿る順番を示す。



(b) トラバース処理手順 2 (部位間の接触検知): 枠内は用いる要素を, (数字) は辿る順番を示す。

図 3 トラバース処理手順

前記判定で接触検知した NE ノードより下位にあるリーフを用いて, 接触した相手の部分木を辿り (物体間の接触を検知することで辿る木構造は限定されている.), 最終的に物体間の接触部位を選出する. 図 3(b)にその探索を示す. 接触部位を選出した後, 力学計算部にその情報を送り, 力学計算部にてペナルティ法により有限要素法に基づき力学計算を行う. この力学計算の計算負荷は高いため, PE ノードで規定されたプロセッサで並行的に行われる. 本論文の主題は高速検知であるので, 力学計算に関しては論じない.

木構造のノード数の増加による接触計算時間の短縮と, 木構造の更新時間とは相反の関係にある. 即ち, 木構造の総ノード数を増やすことで (リーフの BE ノードが小さくなり, また含まれる 4 面体数が少なくなるため), 接触対象となるポリゴン数を削減することができるが, 木構造の更新時間及び木構造を辿る処理時間が増大する. 従って, 用途により適切な分割基準数 K を決める必要がある.

3.3 木構造の更新処理

木構造の更新処理時に, BE ノードをできるだけ小さくすることで, トラバース処理をより高速, かつ精度良く行える. そのため, 物体内部のポリゴンは他のポリゴンと接触しないことを考慮し, BE ノードの更新は, 表面ポリゴンを構成する頂点のみ考慮する.

3.4 処理時間の予測

単純な方法での接触検知の計算量のオーダー (時間のオーダーでもある) は, 対象モデルデータの総ポリゴン数を N としたとき, 総当たりとなるため $O(N^2)$ である.

提案手法の接触検知の計算量 T のオーダーは以下の式により表される.

$$T = O(P \log_2 P) + O\left(\left(\frac{N}{P}\right)^2\right) + O(P) \quad (1)$$

ここで, P は木構造が持つ全ノード数である. 右辺第 1 項が木構造のトラバース処理, 2 項が接触検知処理, 第 3 項が物体の変形に対応する木構造の更新処

理量である. 第 1 項は, 2 進木であるため木の深さが $\log_2 P$ から得られる. 第 2 項は, 葉の BE ノードが持つポリゴン数であり, この総当たりから得られる. また, 第 3 項は更新すべき BE ノード数がほぼ P であることから得た.

また全ノード数 P は以下の式より表される.

$$P = 2 \frac{S}{K} - 1 \quad (2)$$

ここで S は全有限要素数である.

4. 評価

4.1 評価実験

提案手法の評価を行うために実験を行った. 実験に用いた計算機の性能諸元を表 2 に示す. 実験は, 表 3 に示す要素モデルを, 生体モデル記述言語により複合化して作成した 2 ケースの生体モデルデータを用いて (図 4 に示す), その処理時間を計測した.

図 5 に前記データモデルを用いて分割数を変化させた場合の木構造の更新処理時間, トラバース時間, 接触計算処理時間をそれぞれ示す. 図の横軸は木構造のノード数を示し, 縦軸は, ノード数の変化による処理時間 [ms] を示している.

図 4 に示す 2 ケースのモデルデータを用いて, 更新時間, トラバース時間, 接触処理時間の項目に分けて計測した結果を図 5 に示す. また, 図 6 に, 前記 2 ケースのモデルデータの接触判定に必要な全処理時間 (更新時間 + トラバース時間 + 接触計算処理時間) を示す.

4.2 考察

式 (1) で示した概略の計算量と計測データは傾向が一致していることが, 図 5 から分る. 式 (1) 右辺の第 1 項が, トラバース処理時間を示したものであり P を総ノード数としたとき $O(P \log_2 P)$ であるが, この傾向は, ケース 1 および 2 とも図 5(a)(b) に示されてい

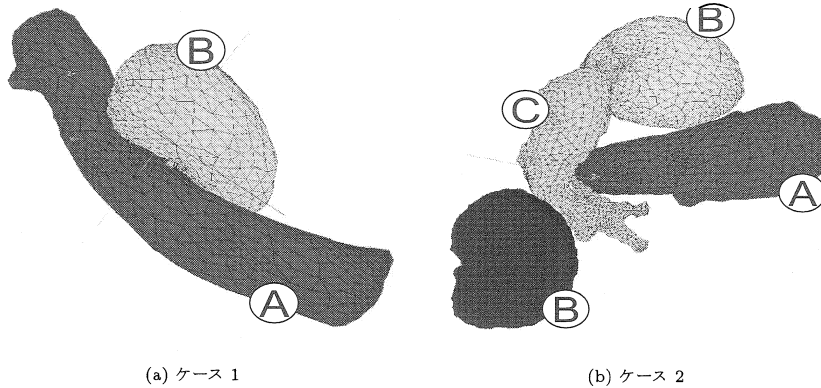


図 4 計測に用いた 2 ケースのモデルデータ: ケース 1, ケース 2 共に腎臓周辺の臓器モデルを示す. 位置関係は現実の人体位置を示していない. 生体モデル記述言語によりそれぞれの部位の拡大・縮小, 移動, 回転後, 複合が可能であることを示している.

表 2 実験に用いたパーソナルコンピュータの性能諸元

番号	機器名	機能性能
(1)	CPU	Intel(R) Core(TM) 2 Quad @2.66GHz
(2)	Memory	RAM: 2.08 GB
(3)	GPU	GeForce8800GTX
(4)	OS	MS WindowXP SP2
(5)	コンパイラ	MS Visual Studio 2005

表 3 実験データを構成する要素となる有限要素モデル

オブジェクト名	節点数	4 面体数	表面ポリゴン数
A (大動脈)	1,597	5,607	2316
B (腎臓)	2,397	9,527	3202
C (大静脈)	2,659	9,780	3958

る. 同様に, 接触計算時間は, N を総ポリゴン数で固定として $O((\frac{N}{P})^2)$ として図 5(c)(d) に示されている. 更新処理時間に関しても, $O(P)$ として, 図 5(e)(f) に示されている. 図 6 から分るように, 計算量は式 (1) 右辺の第 2 項が優位であるので, ほぼ $O((\frac{N}{P})^2)$ である. また, 全合計処理時間であるが, これは, ケース 1 では $P = 19,392 (K = 2)$ の場合が最小であり 5[ms], 同様に, ケース 2 では $P = 68,878 (K = 5)$ が最小で 6[ms] となっている.

5. おわりに

映像発生と力学計算で共用可能な構造化したモデルデータにより, 接触判定計算回数を 5,000-40,000 分の 1 に減らすことが可能となった. これにより, 4 物体 (臓器) 各 2,500 節点に対して, 自己接触を含む臓器間の相互作用による変形の実時間 (30Hz) 模擬が可能となった. また 2 進木のトラバース処理は, 計算対象が削減されるため, 単一の演算装置で実装することが可能となり, 再帰的な処理が実装できるため処理が

簡潔となる利点を持つ.

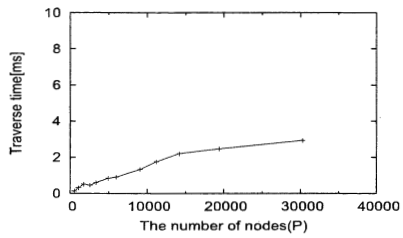
現在, 外力と変位の関係は線形弾性体モデルに基づき計算している. これは, 実時間性からある意味で現実的な対応であるが, 膜の操作等精度の高い手技操作の力学モデルとしては不十分である. 今後, 非線形・異方性の力学モデルと, 提案の接触判定法との親和性を検証する予定である.

謝 辞

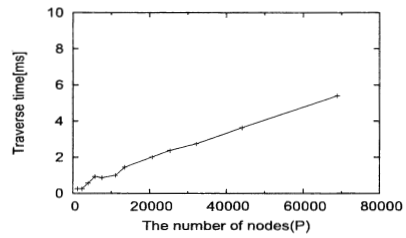
本研究で, ご指導頂いている横浜国立大学 土肥康孝名誉教授, また, データをご提供頂いた横浜市立大学大学院医学研究科 窪田吉信教授, 横山和秀助教に感謝いたします. 本研究の一部は独立行政法人情報通信研究機構 (NICT) 委託研究の下に行われた.

参 考 文 献

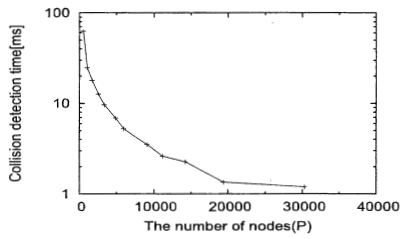
- Ericson, C.: *Real-Time Collision Detection*, Morgan Kaufmann publishers (2005).
- Larsson, T. and Akenine-Möeller, T.: Collision detection for continuously deforming bodies, *Proceedings to Eurographics'01*, pp. 325-333 (2001).
- Nagasaka, M., Takanami, K., Hongo, S., Terada, T., Kikukawa, T. and Ogata, M.: A Real-time Deformation Model using Patient-Specific Medical Data, SIGGRAPH 2007 posters (2007).
- Teschner, M., Heidelberger, B., Müller, M., Pomernets, D. and Gross, M.: Optimized Spatial Hashing for Collision Detection of Deformable Objects, *Proceedings to VMV'03*, pp. 47-54 (2003).
- Yan, J. K.: *Advances in Computer Graphics Imagery for Flight Simulation*, *IEEE Computer Graphics and Applications*, pp. 37-51 (1985).
- 梶原景範: 実時間シミュレータ用地形モデルの生成と表示の一方方法, 第 14 回シミュレーションテクノロジーコンファレンス, pp. 7-13 (1990).
- 緒方正人, 梶原景範, 藤野勝: フライトシミュレータ用高性能ビジュアルシステムの試作, Vol. 93, No. 436, pp. 7-13 (1993).
- 吉隆足立, ステファンホウ, サンジェイシャルマ, 明夫池本, 宜弘熊野: Haptic CAM システムの開発, 精密工学会誌, Vol. 68, No. 4, pp. 591-596 (2002).
- 原田隆宏, 田中正幸, 越塚誠一, 河口洋一郎: グラフィックスハードウェアを用いた個別要素法の高速度化, 日本計算工学会論文集, Vol. 2007 (2007).



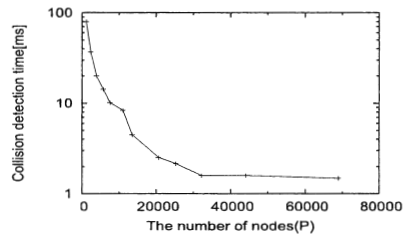
(a) ケース 1: トラバース処理時間



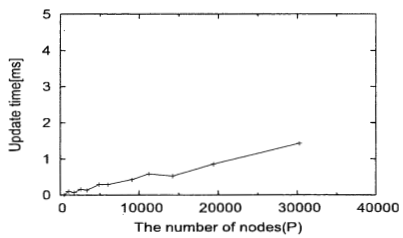
(b) ケース 2: トラバース処理時間



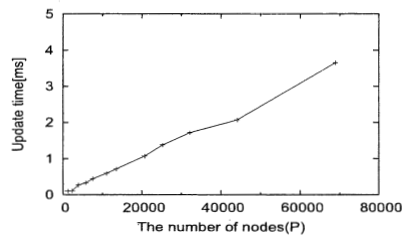
(c) ケース 1: 接触計算処理時間



(d) ケース 2: 接触計算処理時間

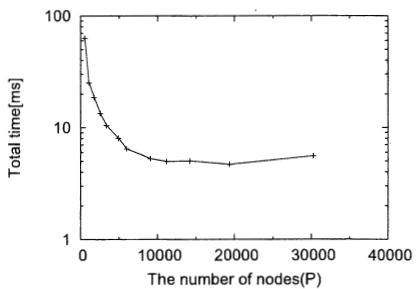


(e) ケース 1: 更新時間時間

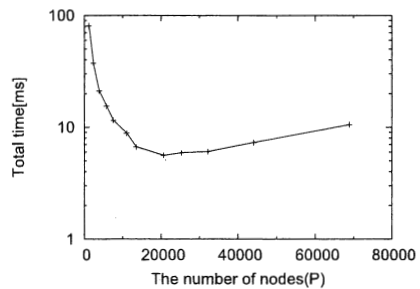


(f) ケース 2: 更新処理時間

図 5 計測結果: 2 ケースのモデルデータを用いて, 更新時間, トラバース処理時間, 接触計算処理時間を計測した.



(a) ケース 1: 合計処理時間



(b) ケース 2: 合計処理時間

図 6 計測結果: 2 ケースのモデルデータを用いて, 全接触判定処理時間を計測した.