

柔らかい仮想物体の力覚表現

広田 光一[†] 金子 豊久[†]

変形する物体の力覚表現においては、有限要素法では計算量が多くなり実時間性が損なわれることから、バネのネットワークによるモデルが一般的に用いられてきた。しかしながら、有限要素法でも線形性と固定境界条件が変わらないという制約のもとでは、全体剛性マトリクスが変形の状態に依存しないことから、この逆行列をあらかじめ求めておくことで、力から変形を比較的容易に計算することが可能である。線形な変形モデルは、大変形では不自然な特性を示すことが知られているが、柔らかさの認識は比較的小さな変形操作で行われることから、本研究では、上述の計算方法を柔らかさの表現に適用することを試みた。指先による変形は一般には物体表面に新たな固定境界条件を生じるが、本手法では、固定境界条件を保存するために、これと等価な力境界条件を与える方法を示した。実際に仮想物体の変形に本手法を適用することで、その有効性を評価した。実装にあたっては、指先と表面との干渉位置の計算に God Object Method を用いた。変形対象として、ノード数 125 および 1713 よりなるモデルを作成し、その柔らかさを力覚提示デバイスを通じて表現できることを確認した。また、各々のモデルで処理に要する時間を計測し、実時間での力覚提示が可能であることを確認した。

A Method of Representing Soft Object in Virtual Environment

KOICHI HIROTA[†] and TOYOHISA KANEKO[†]

In this paper, a method to haptically represent soft object in virtual environment is proposed. The elasticity of the object was defined by using a linear FEM model. By calculating the inverse stiffness matrix in advance, the displacement of nodes according to the affection of force was obtained with low calculation cost. In our prototype system, the displacement of nodes are assumed from the finger position, and the force on nodes that was equivalent to the displacement was approximately calculated. Also, the God Object Method was used to determine the contact point on the object and to calculate the displacement at that point. We applied the method to two objects that consist of 125 and 1713 nodes, respectively. Through this experiment, we measured the calculation time of our method and confirmed that the method is applicable for the real-time presentation of soft object.

1. はじめに

仮想空間における力覚の必要性が指摘され、また、これを表現するためのデバイスに関する様々な研究が行われている¹⁾。このような力覚の表現により提示される情報には様々な側面があるが、最も応用範囲の広い情報として対象物の柔らかさをあげることができる。

柔らかく変形する対象を仮想空間で表現する手法については多くの検討がなされているが、代表的にはバネのネットワークによる表現^{2),3)} (以下、バネモデル)と有限要素法による表現⁴⁾が知られている。一般に有限要素法は計算に多くの時間を要することから、仮想空間のように実時間でのインタラクションを行うシステムでは現実的な手法とはいいがたく、高速な計算機

を利用できる環境で実験的に表現が試みられている⁵⁾にとどまっていた。

一方、バネモデルは、多くの場合バネにより表現された局所的な力-変位関係に基づいてノードを運動させるという差分計算を用いて解かれることから、1サイクルあたりの計算量は比較的少なく、仮想空間における柔らかい物体の表現に多く用いられてきた⁶⁾。しかしながら、このバネモデルを差分法により解く手法も、モデルが複雑になるに従って実時間での計算が困難になる。すなわち、シミュレーション計算における時間と現実における時間とを一致させることが難しくなり、結果として、実時間でのインタラクションにおいては、変形操作に対する対象物の反応が遅く感じられるようになることが問題である。

計算量以外の観点では、有限要素法は材料の変形に関する特性を示す物理的なパラメータを直接記述できるのに対して、バネモデルではこれをモデルの中のバ

[†] 豊橋技術科学大学情報工学系

Department of Information and Computer Sciences,
Toyohashi University of Technology

ネ定数に変換する必要がある。この際、同じ柔らかさの物体を表現する場合でも、バネの接続のジオメトリヤトポロジによって、また、場合によっては個々のバネごとに異なるバネ定数を設定することになる。しかしながら、このようなバネ定数の設定方法については実験的な試みがなされているにとどまっております⁷⁾、確立された手法がないことが問題である。

本研究では、線形な有限要素モデルを用いて物体の柔らかさを実時間で表現することを試みた。「変形の特長」と「柔らかさ」とは同じ意味で用いられることが多いが、ここでは、次のように区別して考えるものとする。すなわち、「柔らかさ」は比較的微小な変形において変形に対する力の変化から認識されるものであり、「変形の特長」のように形状の変化自体に注目したものである。

上述のように、有限要素法による力-変形関係のシミュレーションは、一般には計算量が多く、実時間性を要求される用途には適さないとされる。しかしながら、線形性を仮定することが可能で、かつ、強制変位境界条件が変化しないという状況においては、全体剛性マトリクスがモデルの状態によって変化しないことから、その逆行列をあらかじめ計算しておくことで、変形を比較的少ない計算量で求めることができる⁸⁾。一般に、物体に対する操作では、物体自体が空間に固定されていることから生じる強制変位境界条件に加えて、指先が物体に対して新たな強制変位境界条件を与えると考えられるが、ここではこの強制変位境界条件と等価な力境界条件を求めることで、初期の強制変位境界条件を保存する。以下ではその詳細について述べる。

2. 力-変形関係の表現

有限要素法では対象物全体の変形と力の関係を全体剛性マトリクスにより表現する。これは、要素ごとの力-変位関係を表現する要素剛性マトリクスを要素の集合全体について足しあわせることで得られ、モデルの持つすべての自由度について変位が与えられた際に、各自由度に発生する力を与えるものである。非線形なモデルにおいては、全体剛性マトリクス自体が変位などの関数として定義されるのに対して、線形なモデルにおいてはこれは定数行列となる。以下では、線形なモデルに限った議論を行う。

モデルが強制変位境界条件の与えられる自由度を持つことを考慮すると、全体剛性マトリクスおよびこれにより定義される力-変位関係は次のように与えられる。

$$\begin{pmatrix} F \\ F_0 \end{pmatrix} = \begin{pmatrix} K & K_0 \\ K_1 & K_2 \end{pmatrix} \begin{pmatrix} u \\ u_0 \end{pmatrix} \quad (1)$$

ここに、 (u) および (F) は、力境界条件を与えられたノード（自由ノード）における変位および力、 (u_0) および (F_0) は、強制変位境界条件を与えられたノード（固定ノード）における変位および力である。また、全体剛性マトリクスは、後の議論のために (K, K_0, K_1, K_2) に分解して記述している。

この関係をもとに力境界条件が与えられたノードの変位を求める作業が有限要素法において多くの時間を必要とする部分で、線形なモデルにおいては(自由度) \times (自由度)の大きさの行列の逆変換を計算する必要がある。強制変位境界条件を持つモデルでは、これらの自由度については力が変位に影響しないことから次のような関係の逆変換を求める問題となる。

$$\begin{pmatrix} F \\ u_0 \end{pmatrix} = \begin{pmatrix} K & K_0 \\ 0 & I \end{pmatrix} \begin{pmatrix} u \\ u_0 \end{pmatrix} \quad (2)$$

この解は次のように与えられる。

$$\begin{pmatrix} u \\ u_0 \end{pmatrix} = \begin{pmatrix} L & L_0 \\ 0 & I \end{pmatrix} \begin{pmatrix} F \\ u_0 \end{pmatrix} \quad (3)$$

ここに、

$$L = K^{-1} \quad (4)$$

$$L_0 = -K^{-1}K_0 \quad (5)$$

柔らかさの表現においては、強制変位の境界条件は物体が空間に固定されていることを想定すれば十分であると考えられ、したがって、一般には (u_0) を (0) とおくことができる。以下では、この仮定のもとに議論を行う。この場合、力境界条件を持つ自由度に相当する部分行列にのみ注目することで、これらの自由度の力から変位を計算することが可能である。

$$(u) = (L)(F) \quad (6)$$

特殊な条件として、モデルの中の1つのノード (i) のみに力 (F_i) が作用した場合を考えると、このときのこのノードの変位 (u_i) は次により与えられる。

$$\begin{pmatrix} * \\ u_i \\ * \end{pmatrix} = \begin{pmatrix} | & | & | \\ | & L_{ii} & | \\ | & | & | \end{pmatrix} \begin{pmatrix} 0 \\ F_i \\ 0 \end{pmatrix} \quad (7)$$

同様に、2つのノード (i, j) のみに力 (F_i, F_j) が作用した場合のこれら2点における変位 (u_i, u_j) は次により与えられる。

$$\begin{pmatrix} * \\ \mathbf{u}_i \\ \mathbf{u}_j \\ * \end{pmatrix} = \begin{pmatrix} | & | \\ \hline L_{ii} & L_{ij} \\ \hline L_{ji} & L_{jj} \\ \hline | & | \end{pmatrix} \begin{pmatrix} 0 \\ \mathbf{F}_i \\ \mathbf{F}_j \\ 0 \end{pmatrix} \quad (8)$$

3. 操作にともなう力と変形

現実の世界においては指先が物体にめり込むことはないことから、対象物の変形は指先に従わなければならない。これは、指先が物体に対して新たな強制変位境界条件を与えることを意味する。しかしながら、これを新たに加味して全体剛性マトリクスの逆行列の計算を行うことは計算量の観点からも不適当である。ここでは、この強制変位境界条件と等価な力境界条件を求めることで強制変位に等しい効果を得ることを考えた。

3.1 強制変位に等価な作用力

上述のように、1点のみに力が作用している場合の同一点における変位は力ベクトル (3×1) と部分行列 (3×3) とのかけ算により計算される (式 (7) 参照)。逆に、1点 (i) のみに力が作用しているとした場合に、この点において与えられた変位 (\mathbf{u}_i) を実現する作用力 (\mathbf{F}_i) は、この部分行列 (L_{ii}) の逆行列 (3×3) に変位ベクトル (3×1) を乗することで得られる。

$$\mathbf{F}_i = L_{ii}^{-1} \mathbf{u}_i \quad (9)$$

同様に2点 (i, j) のみに力が作用している場合については以下により計算される (式 (8) 参照)。

$$\begin{pmatrix} \mathbf{F}_i \\ \mathbf{F}_j \end{pmatrix} = \begin{pmatrix} L_{ii} & L_{ij} \\ L_{ji} & L_{jj} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{u}_i \\ \mathbf{u}_j \end{pmatrix} \quad (10)$$

3.2 力の計算

後述のように、本研究では有限要素法の計算に四面要素を用いる。このため、物体の表面は三角形ポリゴンにより表現される。また、指先と物体との干渉もノードの上で生じるとは限らず、一般にはこれらの三角形ポリゴンの上で生じる。したがって、以下では、指先点と対象物との干渉が三角形ポリゴンの上で生じる場合の力の計算法について論じる。

指先がノードの上で干渉している場合には、指先での変位から等価な作用力を式 (9) により計算することができる。しかしながら、辺または面の上で干渉を生じている場合には、単純にこの式を用いることができない。作用力の計算において考慮すべきことは、ノードでの干渉は面との干渉の特殊な場合であると考えられることから、面との干渉にともなって計算される作用力がノードでの干渉にともなって計算される作用力

と矛盾しないこと、また、ポリゴンの境界における力の連続性が求められることから、辺およびノードを介して隣接する三角形ポリゴンとの間で計算される作用力と矛盾しないことである。これらの条件を満たす計算方法として、面積座標を用いた重み付けによる方法を提案する。操作者に提示される力は、干渉を生じている三角形ポリゴンの頂点に位置する3つのノードの間で補間により計算される。このような、力の内挿による計算は、物体形状の滑らかな提示に貢献することが知られており⁹⁾、境界条件の計算方法としてだけでなく力覚提示の観点からも適当な方法であると考えられる。

この方法では、まず、与えられた指先位置での変位 (\mathbf{u}) が、指先と干渉している三角形ポリゴンの3頂点に位置するノード (i, j, k) の各々に単独に作用したと仮定した場合の等価の作用力 ($\mathbf{f}_i, \mathbf{f}_j, \mathbf{f}_k$) を求める。

$$\mathbf{f}_i = L_{ii}^{-1} \mathbf{u}, \quad \mathbf{f}_j = L_{jj}^{-1} \mathbf{u}, \quad \mathbf{f}_k = L_{kk}^{-1} \mathbf{u} \quad (11)$$

これらの力に、指先位置から計算される面積座標の値による重み付けを行うことで、各ノードに与える作用力 ($\mathbf{F}_i, \mathbf{F}_j, \mathbf{F}_k$) を求める。

$$\mathbf{F}_i = \alpha_i \mathbf{f}_i, \quad \mathbf{F}_j = \alpha_j \mathbf{f}_j, \quad \mathbf{F}_k = \alpha_k \mathbf{f}_k \quad (12)$$

なお、面積座標の値 ($\alpha_i, \alpha_j, \alpha_k$) は次により計算される。

$$\begin{aligned} \alpha_i &= \Delta V_j V_j V_k / \Delta V_i V_j V_k \\ \alpha_j &= \Delta V_f V_k V_i / \Delta V_i V_j V_k \\ \alpha_k &= \Delta V_f V_i V_j / \Delta V_i V_j V_k \end{aligned} \quad (13)$$

ここに、 $V_i V_j V_k$ は三角形ポリゴンの頂点の位置、 V_f は指先の干渉位置である。

操作者に返す力 (\mathbf{F}_f) は、ノードに与えられた作用力の総和の反作用として計算される。

$$\mathbf{F}_f = -(\mathbf{F}_i + \mathbf{F}_j + \mathbf{F}_k) \quad (14)$$

3.3 変形の計算

以上により計算された作用力を用いてすべての自由ノードにおける変位を計算することで、物体全体の変形を計算する。線形なモデルでは変位は力について線形であり、複数のノードに力が作用した場合の変位は、単独のノードに力が作用している状態での変位の和として計算される。特に、上述のように3点 (i, j, k) に力 ($\mathbf{F}_i, \mathbf{F}_j, \mathbf{F}_k$) が作用している場合のすべてのノードの変位は次で与えられる。

$$\begin{pmatrix} \mathbf{u} \end{pmatrix} = \begin{pmatrix} | & | & | \\ \hline L_i L_j L_k \\ \hline | & | & | \end{pmatrix} \begin{pmatrix} 0 \\ \mathbf{F}_i \\ \mathbf{F}_j \\ \mathbf{F}_k \\ 0 \end{pmatrix} \quad (15)$$

4. 仮想空間における実装

実装にあたっては、以上で述べた力および変形の計算に先立ち、指先と対象物とのポリゴンの上での干渉点を特定する処理が必要となる。ここでは、God Object Method¹⁰⁾によりこれを実現した。この方法は、指先が物体の内部にめり込んだ場合に、仮想の指先（これを God Object と呼ぶ）を定義し、これを物体表面にとどまらせ、これが表面の制約を満たしつつ指先に追従することをシミュレートする方法である。この方法の利用により、ポリゴン上の指先位置ベクトルと、この点に対する指先の変位ベクトルを得ることができる。なお、この God Object Method を、変形前の形状に対して適用する方法と、変形状態のモデルに対して適用する方法とが考えられるが、ここでは前者の方法をとった。理由として、後者の方法では、対象物の持つエッジなどが移動した際の指先が引っかかる位置を正確にシミュレートできるなどの利点を持つ反面、God Object Method に用いられている表面をたどるアルゴリズムを不安定にし、振動などを生じる場合があるなどの問題を持つことがあげられる。また、柔らかさの表現の観点からは、多くの場合は微小な変位での表現のみで十分であり、この場合、変形前後の形状に大きな違いを生じていないと考えられるためである。

変形の計算および結果の描画は DOS/V PC (Pentium Pro 200 MHz × 2, Windows NT) により行われた。描画の高速化のためにアクセラレータ付きのビデオカード (FireGL 1000, Diamond Multimedia) を使用した。変形の操作点の入力と力覚の提示を行うデバイスとして PHANToM (SensAble Technologies)¹¹⁾を用いた (図1)。また、ソフトウェアとして PHANToM Basic I/O ライブラリを利用した。このライブラリは、力覚提示の処理と視覚提示の処理とを別のスレッドに分けて並列に実行し、特に力覚提

示の処理を 1 [kHz] の周期で呼び出す機能を提供する。

力覚の提示においては力の更新の周波数を高くとることが望ましい¹²⁾。上述の力の計算と変形の計算とは、前者で求められた力境界条件を後者の計算で参照する以外は相互に独立である。この性質をこの点を利用して、力の計算 (式 (11)~(14)) は力覚提示の処理のスレッドに、変形の計算 (式 (15)) を視覚提示の処理のスレッドに割り当てる。これにより、力の更新の周期を視覚提示の処理の負荷に依存しないものとすることができる。

一方、変形する対象物としては、立方体形状 (1 辺 50 [mm]. 以下, 立方体) およびネズミの形状 (体長約 120 [mm]. 以下, ネズミ) の 2 種類を用いた。これらのモデルの複雑さおよび物理的特性を表 1 に示す。立方体は、アルゴリズム的に生成されたもので、この形状を $4 \times 4 \times 4$ に分割した立方体をさらに 5 個の四面体に分割することで、四面体要素による有限要素モデルを作成した。ネズミはぬいぐるみの形状を三次元形状計測装置 (Model3030MM, Cyberware) により入力したもので、四面体要素への分割は手作業により行った。

これらの物体と指先との干渉は物体の表面のみで発生する。したがって、上述の計算で作用力を受けるのは表面のノードに限定される。また、視覚表現では、一般には表面のみが描かれることから、変形の計算についても、表面のノードのみについて行えば十分である。これらのことから、ノードのデータは表面のノードのみについて保持していればよく、また、逆剛性マトリクスは、表面のノードに関する部分のみを保持していればよい。この逆剛性マトリクスに要するメモリ上のデータ領域は、表面の自由ノードの数の自乗に比例し、立方体では約 384 kbyte ($= (73 [\text{node}] \times 3 [\text{d.o.f.}]^2 \times 8 [\text{byte}])$)、ネズミでは約 77 Mbyte ($= (1036 [\text{node}] \times 3 [\text{d.o.f.}]^2 \times 8 [\text{byte}])$) となっている (ともに倍精度浮動小数で定義)。

立方体のメッシュの構造およびこれに対する操作の様子を図 2 に示す。なお、変形時の図中には、ポイン

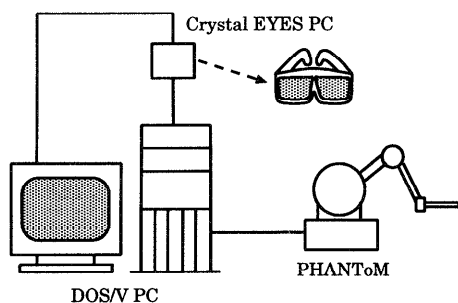
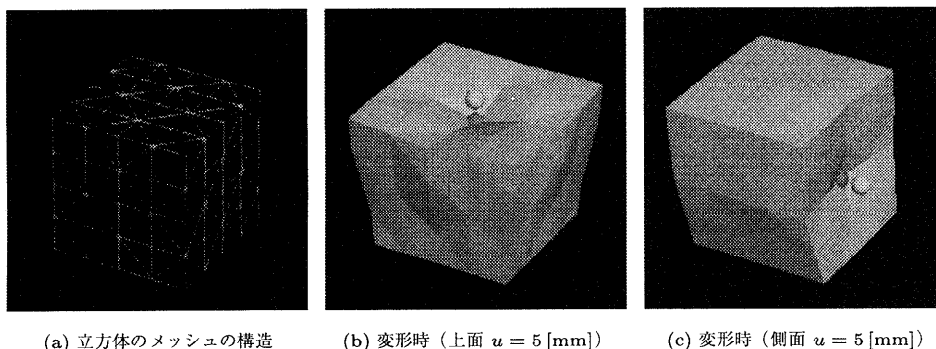


図1 実験装置の構成

Fig. 1 System block diagram.

表 1 変形に使用したモデル
Table 1 Specification of models.

	立方体	ネズミ
総ノード数	125	1713
表面ノード数	98	1175
表面自由ノード数	73	1036
表面三角形パッチ数	192	2346
四面体要素数	320	6312
ヤング率 [Pa]	10000	1000
ポアソン比 [-]	0.49	0.3

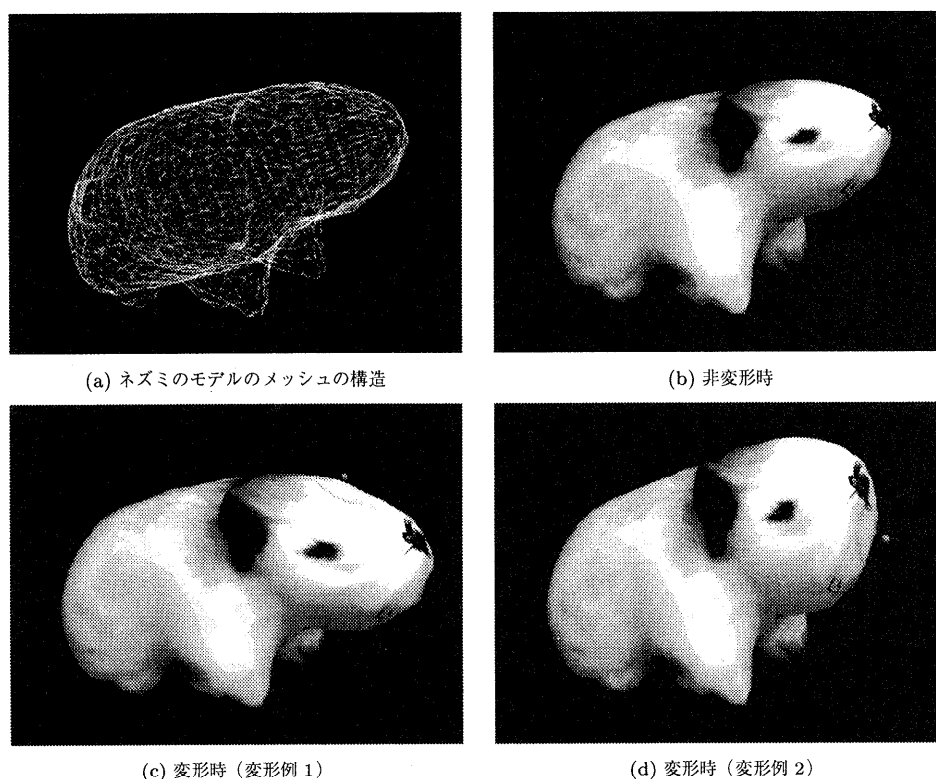


(a) 立方体のメッシュの構造

(b) 変形時 (上面 $u = 5$ [mm])(c) 変形時 (側面 $u = 5$ [mm])

図 2 変形例 (立方体)

Fig. 2 Example of deformed shapes (cube).



(a) ネズミのモデルのメッシュの構造

(b) 非変形時

(c) 変形時 (変形例 1)

(d) 変形時 (変形例 2)

図 3 変形例 (ネズミ)

Fig. 3 Example of deformed shapes (mouse).

タおよび God Object の位置が小球により示されている。上述のように、God Object は非変形時の物体の表面に拘束されていることが確認される。

より複雑な物体の変形例として、ネズミのモデルに対する操作の様子を図 3 に示す。立方体のモデルと比較してはるかに複雑であるが、これについても実時間での変形表現が可能であることが確認された。なお、計算処理に要する時間に関しては 6 章で論じる。

5. 大変形時の視覚表現の改善

本章では、上述のアルゴリズムで生じる比較的大きな変形の際の視覚的な不自然さを修正する方法について述べる。この視覚的な不自然さは、線形なモデルが、変形を生じていない状態近傍での微小な変形における力-変位関係を定義しており、したがって、大きな変形においては誤差が顕著となることによる。このよう

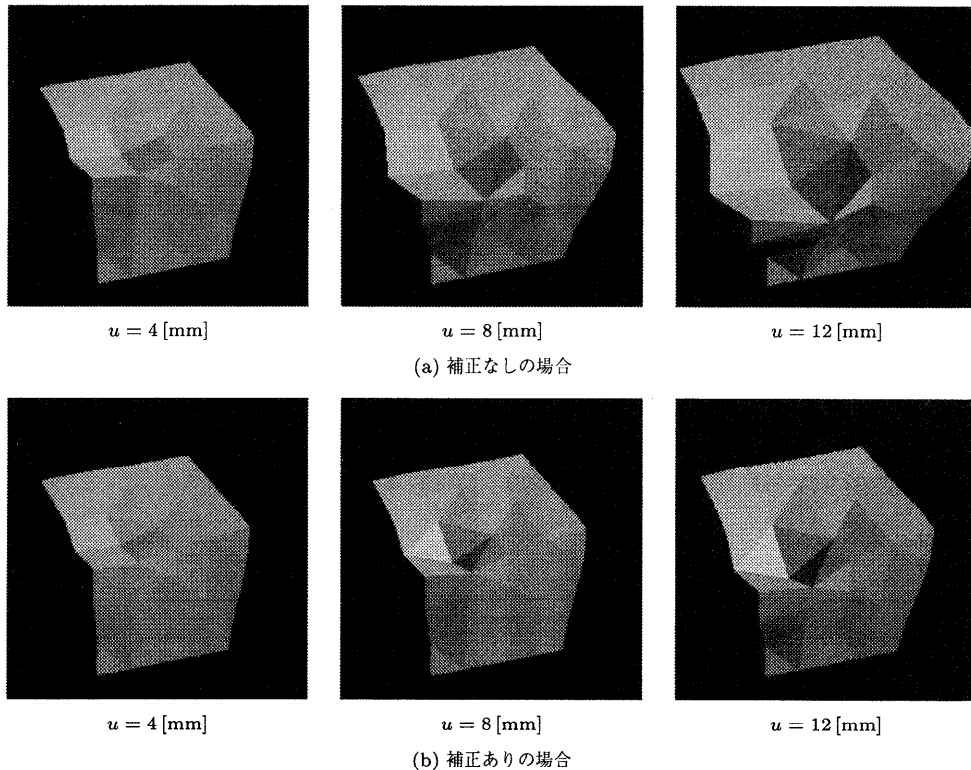


図4 変形の比較
Fig. 4 Effect of visual correction.

な問題の1つの解決方法は、変形の大きさに応じて剛性マトリクスを使い分けることである。非線形なモデルによる有限要素法では変形状態の変化に応じて、その状態の近傍での力-変形関係を定義する接線剛性マトリクスを計算することが行われるが、この方法では、変形を求めるのに先立ち接線剛性マトリクスの逆行列を計算する必要が生じ、実時間の用途には適さない。

大きな変形の際の形状を自然なものとする1つの方法として、大変形時にこれに適した逆剛性マトリクスを用いることが考えられる。逆剛性マトリクスは、各々の自由度に微小な力が作用した際の変位をすべての自由度について求め、これをその力の大きさに除したものと解釈することができる。したがって、大変形に適した逆剛性マトリクスは、大変形時の変位をそのときの力で除した近似の逆剛性を、すべての自由度について計算することでこれを求めることができる。

以下の実験では、幾何学的非線形を考慮した有限要素モデルを用いたシミュレーションにより近似の逆剛性の値を求めた。モデルの要素分割は線形な場合と同様とし、ニュートン法による繰返し計算でモデルの変形を計算する。具体的には、モデルの各ノードの各自

由度 (j) に対して単独に $\pm F_0$ の力を作用した場合のすべての自由度 (i) の変位 $u_{ij}^{\pm F_0}$ を求め、これを用いて、近似の逆剛性の値 $L_{ij}^{\pm F_0}$ を $u_{ij}^{\pm F_0} / (\pm F_0)$ により計算した。これは、 F_0 の力で生じる程度の変形を大変形と呼ぶものと仮定したことを意味する。結果として、各自由度について逆剛性 L_{ij}^0 に加えて、近似の逆剛性 $L_{ij}^{+F_0}$ および $L_{ij}^{-F_0}$ が得られる。これらを用いて、力から変位を次の式により計算する。

$$u_i = \begin{cases} \left\{ \left(1 - \frac{F_j}{F_0} \right) L_{ij}^0 + \frac{F_j}{F_0} L_{ij}^{+F_0} \right\} F_j & (F_j > 0) \\ \left\{ \left(1 + \frac{F_j}{F_0} \right) L_{ij}^0 - \frac{F_j}{F_0} L_{ij}^{-F_0} \right\} F_j & (F_j \leq 0) \end{cases} \quad (16)$$

以上の手法により特性の補正を行った場合の変形と、補正を行わなかった場合の結果とを比較した。同一のノードに対して同一の変位を与えた場合の変形の様子を図4に示す。なお、この例では F_0 の値は $2[\text{N}]$ としている。一般に、線形モデルでは、微小変形時の変形を外挿するために不自然となっていたが、補正を行うことでこれが改善されることが確認された。ただし、この計算方法では線形の場合の逆剛性の値 (L^0) に加

えて、大変形時の値 (L^{+F_0}, L^{-F_0}) を保持する必要があることから、補正を行わない場合の3倍のデータ領域を必要とする。

6. 計算時間の評価

力覚の提示をともなう仮想空間での利用を想定した柔らかさの表現においては、力および変形の計算を実時間で行う必要がある。本章では、上述の手法の計算時間について評価した。上述のように、計算は変位から等価な作用力を求める計算（力の計算）と、作用力に基づいて変形を求める計算（変形の計算）とに分けて考えることができる。これらの値を、立方体（大変形時の補正なし、あり）およびネズミのモデルに対する操作について実測した。また、実際の操作に影響をあたえる描画のレートについても参考のために示した（表2）。なお、表中の描画レートは1秒あたりの描画関数の呼び出し回数を調べたもので、立方体についてはポリゴンによる描画（図2(b), (c)参照）を行っている。また、ネズミの描画レートはテクスチャマッピングを行った場合（図3(b), (c), (d)参照）の値であり、ワイヤフレームおよびテクスチャマッピングを行わないポリゴン描画では6.8 [Hz]であった。

上述のように、力の計算と変形の計算とは、前者で求められた力の値を後者が参照するという独立性の高い関係にある。したがって、力の計算を含む力覚提示の処理と、変形の計算を含む視覚提示の処理とを、独立のループで異なる時間間隔で実行することが可能である。一般に、力覚提示の処理は1 [kHz]以上のサイクルで行うことが望ましい¹²⁾とされる。また、描画はビデオのフレームレートである60 [Hz]程度で行うことが望まれる。計測結果は、上述の手法が力覚提示および視覚提示のいずれについても、これらの要求を満たす程度の時間で処理できることを示している。なお、力の計算は理論的にはモデルの複雑さに依存しないはずであり、表に示した結果のばらつきは、計測時の計算機の負荷の状態などによるものであると考えられる。一方、変形の計算はモデルの自由ノードの数に比例することが予想され、実際、立方体（補正なし）とネズミとではおおむねこの関係を満たした結果となっ

ている。

7. おわりに

本研究では、物体の比較的小さな変形操作における柔らかさの提示について検討し、線形な有限要素モデルを用いてこれを実時間で表現する手法を提案した。この手法は、物体の変形とこれにともなう力を直接計算により求めることを特徴とし、これにより、パネモデルを用いた差分計算のような繰り返し計算による手法では得ることの難しい速い変形を表現することが可能である。

今後の課題の1つとして、逆剛性のデータの持ち方に関する検討があげられる。本研究で提案した手法の1つの問題は、逆剛性の値をすべての自由ノードの組合せに対して持つことから、これを格納するために多くのメモリ領域を必要とすることである。たとえば、ネズミのモデルは、仮想物体として必ずしも複雑な形状とはいえないが、その変形の計算のために、上述のようにおよそ77 MBのデータをメモリ上に保持している。実際には、近くにあるノードの力-変位関係は互いに類似していることが予想されることから、ノードの空間的な位置関係に基づく補間により、より少ない点での逆剛性のデータから各点での値を計算することが考えられる。

もう1つの課題として、接触面積を考慮した力の計算があげられる。本稿で述べた力および変形の計算では、指先を点と見なし、これと三角形ポリゴンとの接触を仮定していた。有限要素法におけるノードへの集中的な荷重は、これに接続している表面ポリゴンへの分布荷重を意味する。したがって、体積要素の大きさ（すなわち表面ポリゴンの大きさ）の変化は、荷重を想定している分布面積の変化を意味する。具体的には、同一の形状であっても、表面ポリゴンの細かさが変化した場合に、変形の鋭さが変化し、したがって、提示される力も異なったものとなる。比較的広い面積で指先と対象とが接触するような状況については、理論的には接触している範囲近傍のすべてのノードへの力の作用を仮定することで計算できると考えられるが、関与するノードの増加にともなって計算量が増大する。

筆者らはまた、提案した手法を2指により対象物をつまむ操作に適用する方法について検討したいと考えている。このための1つの課題は、指先による強制変位境界条件が2点に対して与えられた場合の、等価な作用力の計算方法にある。これらの指先による境界条件がともにノードに対して作用している場合は、等価な作用力を式(10)により求めることができ、実際に

表2 計算時間の評価
Table 2 Calculation time.

	立方体	立方体 (補正)	ネズミ
力の計算 [μ s]	14.9	10.8	12.1
変形の計算 [μ s]	157.5	702.5	4182.5
描画レート [Hz]	97.3	97.3	1.9

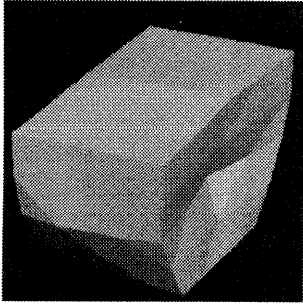


図5 変形例 (立方体, 独立2点)
Fig. 5 Example of a deformed shape
(cube, independent two nodes).

図5に示すような変形を得ることができる。この図では、左右の側面の中心にあるノードに左右から挟み込むように変位を与える操作を行っている。しかしながら、一般には指先は三角形ポリゴンの上で干渉することから、この計算方法を、1点の場合と同様に、力覚的に矛盾なく拡張する必要がある。このような問題の解決方法について、今後検討していく予定である。

参考文献

- 1) 廣瀬通孝：バーチャル・リアリティ，産業図書(1993)。
- 2) Norton, A., Turk, G., Bacon, B., Gerth, J. and Sweeney, P.: Animation of fracture by physical modeling, *Visual Computer*, Vol.7, pp.210-219 (1991).
- 3) 矢野博明, 岩田洋夫：自律的自由曲面を用いた仮想環境による協調作業, 電学論C, Vol.115, No.2, pp.215-222 (1995)。
- 4) Terzopoulos, D., Platt, J., Barr, A. and Fleischer, K.: Elastically Deformable Models, *Computer Graphics*, Vol.21, No.4, pp.205-214 (1987)。
- 5) Ogi, T., Hirose, M., Watanabe, H. and Kakehi, N.: Real-time Numerical Simulation in Haptic Environment, *Proc. HCI '97*, pp.965-968 (1997)。
- 6) 広田光一, 高橋英嗣, 金子豊久, 関口隆三, 森山紀之：手術シミュレーションのための柔らかい臓器のモデルと操作環境, 医用画像工学会誌, Vol.16, No.1, pp.31-42 (1998)。
- 7) 広田光一, 金子豊久：仮想物体の弾性モデルに

関する検討, 計測自動制御学会論文集, Vol.34, No.3, pp.232-238 (1998)。

- 8) 矢川元基, 吉村 忍：計算力学とCAEシリーズ1：有限要素法, 培風館(1991)。
- 9) Morgenbesser, H. and Srinivasan, M.: Force shading for haptic shape perception, *Proc. ASME Dynamic Systems and Control Division*, DSC-Vol.58, pp.407-412 (1996)。
- 10) Zilles, C. and Salisbury, K.: A Constraint-Based God Object Method for Haptic Display, *Proc. IROS '95* (1995)。
- 11) Massie, T.: Initial Haptic Explorations with the Phantom: Virtual Touch Through Point Interaction, Master's Thesis, M.I.T. (1996)。
- 12) Shimoga, K.: A study of perceptual feedback issues in dextrous tele-manipulation: Part I, Finger force feedback, *Proc. VRAIS '93*, pp.263-270 (1993)。

(平成10年6月24日受付)

(平成10年10月2日採録)



広田 光一 (正会員)

1988年東京大学工学部産業機械工学科卒業, 1990年同大学院産業機械工学専攻修士課程修了, 1994年同博士課程修了, 1995年豊橋技術科学大学情報工学系助手。主にヒューマンインタフェースの研究に従事。日本VR学会, 日本機械学会, 計測自動制御学会各会員。



金子 豊久 (正会員)

1962年東京大学工学部電気工学科卒業, 1964年同大学院電気工学専攻修士課程修了, 1970年プリンストン大学電気工学科博士課程修了(Ph.D.), 同年IBM入社ワトソン研究所研究員, 1982年日本IBM東京基礎研究所に移り, 東京サイエンティフィックセンター長等を歴任, 1994年豊橋技術科学大学情報工学系教授。主に医用画像処理とCGの応用に関する研究に従事。ACM, IEEE, 電子情報通信学会各会員。