

力覚提示を目的とした刃先と対象物との干渉計算

田中厚子[†] 広田光一[†] 金子豊久[†]

切断操作においては刃先と対象物との干渉により刃先に力が生じる。切断操作にともなう力覚の提示を仮想空間で実現するためには、仮想物体と刃先との干渉の状態をシミュレートする必要がある。従来の干渉計算は物体表面との干渉に重点がおかれていたが、切断操作においては刃先と物体内部の干渉計算が必要となる。本研究では、仮想物体の表面および内部での刃先の移動を、より一般的に力学的な拘束のある空間と指先との干渉の問題ととらえてモデル化する。力覚の提示は、拘束条件に厳密に従う理想の指先を定義し、実際の指先の移動にともなう理想の指先の移動の処理と、両者の乖離に基づく拘束力の計算により実現される。干渉の状態は空間を占める材料によって異なることから、空間を四面体形状の部分領域に分割し、各々に材料を定義する。また、領域内部での力学的な規則に基づく指先の移動の計算、領域境界での拘束状態の定義、状態遷移による領域間の移動などの考え方により、指先の移動と指先が属している要素を検索する。以上の計算手法を実装して処理時間を計測することで、力覚提示に利用できる程度の実時間性が得られることを確認した。また、仮想空間での刃先と物体との干渉計算に適用することで力覚の提示を実現した。

Simulation of Constrained Motion of Cutting Edge in Virtual Environment

ATSUKO TANAKA,[†] KOICHI HIROTA[†] and TOYOHISA KANEKO[†]

In this paper, we propose a method of simulating the motion of cutting edge in virtual environment. We abstract the problem as the constrained motion fingertip in space. It is a common approach to simulate the colliding force by defining an ideal fingertip that moves while satisfying physical constraints. In the approach, the constraint force is obtained from the disparity between the real and ideal fingertips. In our study, we propose a method of simulating the spatial movement of the ideal fingertip. The constraint forced on the fingertip differs according to the material that occupies the space. To represent spatial variation of the material, we divide the space into tetrahedral elements and defined the characteristics of the element. The movement of fingertip in an element is simulated on the basis of physical constraint. Fingertip transfer between neighboring elements and the change of bound state on the boundary is described by introducing the idea of state transition. We implemente the proposed method, estimated the computation time, and proved that the method is fast enough for the application to real-time simulation. Also, we introduced the method to the representation of cutting force in a virtual environment.

1. はじめに

計算機によって作り出された合成情報を人間の感覚器官に直接提示することにより、仮想世界の能動的な疑似体験を可能にする仮想現実感 (Virtual Reality) の技術が注目を集めている。仮想空間の実現にあたっては、視覚・聴覚の提示に加え触覚・力覚を含む体性感覚の提示が重要であるとされる^{1),2)}。

力覚表現においては、指先と物体との干渉の計算は重要な問題の1つである。ここに、指先とは操作点と

物体の干渉点を示すものである。物体との接触にともなう力覚情報の提示では、指先が物体との接触により拘束されている位置を仮定し、この点と指先との距離に基づいて反力を計算する手法が一般的に用いられている³⁾。このような力提示の枠組みにおいては、指先の位置の変化に基づいて拘束点の位置を移動することで、たとえば表面をなぞるなどの操作にともなう力覚の表現が可能となる。なお、干渉計算という用語は衝突判定という意味で使われることもあるが、本論文では上述のように物体と指先との間で生じる拘束関係とこれに基づく拘束点の移動に関する計算という意味で使用される。また、以下の議論では指先を点として定義する。現実の世界では指先は大きさを持つが、力覚提

[†] 豊橋技術科学大学情報工学系

Department of Information and Computer Sciences,
Toyohashi University of Technology

示においてはこれを点として扱うことが多く、一般性を損なうものではない。

力覚提示を目的とした干渉計算においては、計算の実時間性、力学的規則に基づく干渉点移動の計算の必要性、指先移動の連続性の考慮が必要とされる。実時間性は、提示力を指先移動に対して現実感を損なわない程度に短いサイクルで更新する必要があることを意味する。力覚の提示においては、使用するデバイスの特性にも依存するが、一般には 1 [kHz] 程度のサイクルでの力の更新が望ましいとされる⁴⁾。一方、現実の現象を模した力覚提示においては、拘束点の移動の条件は物理的な規則として定義される。したがって、干渉計算ではこの規則を満たす拘束点の移動を行う必要がある。指先移動の連続性とは、指先の移動の経路を意識する必要があることを意味する。仮想空間システムでは一般に指先位置が時間に関して離散的にサンプリングされることから、その間の移動の経路について何らかの仮定をおいた計算が必要である。

従来の衝突判定のアルゴリズムにおいては、物体相互の交差の発見に重点がおかれていた⁵⁾。本研究では上述のように点と物体との関係を扱うという意味で興味を異にする。これらの手法で用いられている、オクトツリーによる領域分割や距離をもとにした交差判定の省略の手法自体は本研究の目的とする干渉計算にも適用の可能性がある。ただし、上述のような干渉計算における要求を満たすためには、これらの高速化手法の導入のみでは不十分であると考えられる。

力覚提示を目的とした物体表面と指先との干渉の計算方法としては God-object Method が知られている⁶⁾。これは、物体表面での拘束を考慮した理想の指先 (God-object) の位置を、現在と過去の指先位置から一意に決定する手法で、とくにポリゴンにより定義された表面形状と指先との干渉については実時間性の高い計算が可能である。また、God-object Method を拡張し、線分とポリゴン面との干渉計算を行う試みもなされている⁷⁾。この方法では、操作者の持つ道具である直線を線分と両端点から構成されるものとし、点と点、点と直線、直線と直線、点とポリゴンという 4 種類の干渉のみを対象とすることで計算を単純化している。これらはいずれも、物体表面との干渉に注目したものであり、物体内部との干渉は考慮されていない。

筆者らの関心は、切断操作における反力の提示を目的とした刃先と対象物との干渉計算にある。これまで刃先と対象物との干渉計算にはボクセルモデルが広く利用されてきた。ボクセルモデルでは、ボクセルごとにこれが占める空間の性質を定義し、刃先位置のボク

セルの性質を参照することで、物体と刃先との接触やこれにともなう力を計算する。ボクセルによる表現では、刃先位置のボクセルの参照にともなう計算負荷が少なく、また、操作の観点からは道具（すなわち、刃先など）と交差するボクセルを消去するという簡単な操作で切断や切削が実現される利点がある。このようなボクセルモデルの特性を手術シミュレーション⁸⁾や仮想彫刻^{9),10)}に応用する試みがなされている。ただし、この手術シミュレーションの試みでは力覚は考慮されておらず、仮想彫刻の試みにおいても道具の速度に比例した力が表現されるという単純化された力覚提示が行われているにとどまっている。一方、ボクセルによる表現の問題として、空間の境界の滑らかな表現が難しいことがあげられる。このことは、指先を物体表面に拘束することで表面をなぞるなどの操作にはボクセルによる表現が適さないことを意味する。

切断にともなう刃先への作用力の表現に関しては、簡略化されたモデルによる計算手法が提案されている¹¹⁾。この中では、実際の刃先位置に加えて刃先が物体の中をどこまで切り進んでいるかを示す非変形時の刃先を定義することで、刃先に作用する力が計算され、また、刃先を離散化することで力の分布が計算される。非変形時の刃先は God-object Method における理想の指先に相当するもので、これを力学的規則に従って移動することで切断の進行が表現される。このような切断操作は、物体の表面ではなく物体が占める空間と刃先との干渉と考えることができる。同様に、粘性を持つ流体を棒でかき回す操作なども広い意味での指先と空間との干渉にともなう力を発生する現象と抽象化することができる。したがって、以下では、刃先についても指先という呼び方を用いる。

指先と空間との干渉の基本的な考え方は、空間を部分空間に分割し、各々の部分空間に性質を定義し、指先が属している空間の性質を参照することで、指先と空間との相互作用を計算することにある。ボクセルによる表現では、部分空間を直方体領域の規則的な配列として定義していることが上述のような表面の粗さなどの問題を生じていた。部分空間の定義という意味では、これを領域としてではなく境界により規定する方法も考えられる。これは、具体的には上述の物体表面との干渉計算により領域の出入りを検出することで指先がどの部分空間に存在するかを特定するというアプローチの可能性を示している。本研究では、空間を部分空間 (四面体要素) の集合として表現し、この表現に基づいて干渉計算を行う手法を提案する。これは、部分空間の境界をポリゴンモデルにより定義する方法

表 1 状態遷移
Table 1 State transition.

現在の状態	遷移後の状態								
	現在属する四面体				隣接する四面体				外部
	四面体	面	辺	点	四面体	面	辺	点	
四面体			x	x	x	x	x	x	x
面				x		x	x	x	
辺							x	x	
点								x	
外	x	*	x	x	x	x	x	x	

* : 外部からは空間を構成する四面体の表面に遷移

の一種ではあるが、部分空間の形状を四面体に限定することにより、指先が領域を出入りする際に交差する面が限られ、干渉計算が簡単になるという利点がある。また、空間の形状を四面体要素の集合により表現することは、空間の境界が三角形ポリゴンで記述されることを意味し、ボクセルと比較して滑らかな表面形状の表現が可能である。一方、指先の移動にともない指先が属している要素を検索するアルゴリズムが必要となる。以下にその詳細を論じる。

2. 干渉計算の実現

本手法では、指先位置の移動から材料の性質を考慮しつつ理想の指先位置を計算する。ここに理想の指先とは、本来指先があるべき位置を示すものである。たとえば、進入することのできない材料の表面を指先が通過した場合、指先は表面上に拘束されているはずである。したがって、実際の指先が進入しても、理想の指先は表面に拘束される。材料の性質の考慮には、進入の許されない領域表面での拘束や、摩擦力による指先の固定などが含まれる。これは、空間に定義されている性質のもとで、理想の指先に許される移動を計算することで実現される。本研究では、上述のように空間を四面体要素の集合として表現する。また、材料の性質、現在の指先および理想の指先位置が与えられると、移動後の理想の指先位置が計算されるものとする。この移動の計算は、本来干渉計算ではなく力覚表現の手法に依存する部分であることから、力学的モデルにより与えられるものと仮定する。すなわち、このモデルを参照して理想の指先の移動を計算することにより、力学的規則を満たした移動を実現する。

干渉計算の実現にあたっては、要素への進入の可能性の確認と拘束状態の導入という2つの考え方をを用いた。前者では、要素の境界を越える場合には、一度境界まで移動し、隣接する要素の材料の性質に従ってその内部に移動できるかどうかの判定を行う。進入できない場合には、理想の指先は境界に固定される。こ

れにより、本来進入の許されない領域を飛び越えるなどの不都合が回避され、指先の移動の連続性が保証される。ただし、指先のサンプル点の間の軌跡は直線とする。後者では、要素の中で指先がとりうる状態として四面体拘束、面拘束、辺拘束、点拘束を考える。それぞれの拘束状態で理想の指先が移動しうる範囲は、四面体要素の内部、三角形平面、線分、点である。また、これらの状態の間の変化を状態遷移の考え方で記述し、遷移の条件を定義する。アルゴリズムでは、隣接要素への移動と要素内での状態の変化をいずれも状態遷移により実現する。なお、実現においては上述の状態に加えて、理想の指先がいずれの四面体にも属さない外部状態も考慮する。状態遷移表を表1に示す。

手続きとしては、まず、現在の状態から自由度を拡大する遷移(表1中の)が可能であるかを調べる。自由度が拡大できない場合は、現在の拘束状態での移動を行う(同表中の)。拘束状態で制限されている範囲を越えた移動を生じた場合には、領域の境界に相当するより自由度の少ない状態に遷移する。たとえば、四面体内部から隣接する四面体の内部への移動は、現在拘束されている四面体内での移動による面拘束状態への移行と、その面から隣接する四面体へ自由度拡大という2回の状態遷移で表される。拘束状態を仮定することの利点は、拘束要素が平面や直線のように幾何学的に単純であることから、拘束状態での運動を計算しやすいことがあげられる。

アルゴリズムの詳細の説明に先だち、説明の中で共通に使用する記号を定義する。まず、現在の指先位置を p 、現在の理想の指先の位置を q とする。理想の指先は、つねに実際の指先の方向に移動を試みる。しかしながら、理想の指先が面および辺に拘束されている状態では、必ずしも実際の指先の方向に移動できるわけではない。そこで、理想の指先の拘束状態を考慮して、移動可能な位置に目標の指先 p' を定義する。具体的には理想の指先を拘束する面または辺に実際の指先 p から下ろした垂線を目標の指先とする。これは、

理想の指先と実際の指先との距離に比例した線形な作用力を仮定するモデルでは、 $p' - q$ より得られる力が $p - q$ より計算される力の拘束自由度方向の成分に等しいことによる。理想の指先の移動は、上述のようにモデルに与えられているものとする。アルゴリズムの中では、その計算を材料の性質 id 、指先の目標位置 p' 、現在の理想の指先位置 q より、移動後の位置 q' 、および移動の可否を与える次のような移動関数により行うことができるものとする。以下では、この移動関数を用いて各拘束状態のアルゴリズムを述べる。アルゴリズムは、自由度を拡大する遷移と現在の拘束状態での移動に分けて説明する。

$$\text{move}(id, p', q, q') = \begin{cases} 0 & \text{移動しない} \\ 1 & q' \text{へ移動する} \end{cases} \quad (1)$$

2.1 四面体拘束

(a) 自由度拡大

四面体拘束からの自由度拡大はない。

(b) 現在の拘束状態での移動

- (b-1) $\text{move}(id, p', q, q')$ により q が移動するか調べる。ここに、 id は q が現在属する四面体の材料、3自由度の移動が許されることから p' は p に等しい。移動しなければ、現在の拘束状態を維持する (q は不変)。
- (b-2) q' が四面体の範囲内であれば現在の拘束状態を維持する。現在拘束されている四面体の4面のそれぞれ (i) について、面上の1点を o_i 、面の法線を n_i とすると次式を満たしていれば q' は外部と判定される。

$$(q' - o_i) \cdot n_i \geq 0 \quad (2)$$

すべての面について内部と判定された場合には、現在の拘束状態を維持する ($q = q'$)。

- (b-3) q' が四面体の範囲外であれば自由度を縮小する。現在拘束されている四面体の4面のそれぞれ (i) について、面上の1点を o_i 、面の法線を n_i とし、面 i と線分 (q, q') との交点 c_i を次式により求める (図1)。

$$c_i = \frac{d_0 q' - d_1 q}{d_0 - d_1} \quad (3)$$

ここに、

$$\begin{aligned} d_0 &= (q - o_i) \cdot n_i \\ d_1 &= (q' - o_i) \cdot n_i \end{aligned} \quad (4)$$

ただし、 $d_0 \cdot d_1 > 0$ の場合は、交点は線分 (q, q') 上にないことから、 c_i は無効である。交点が見つかった場合には、距離 $|q - c_i|$ を最小にする c_i を選択し、面拘束状態に遷移する ($q = c_i$)。

2.2 面拘束

(a) 自由度拡大

- (a-1) 移動方向が外向きかどうかを、現在属する面の

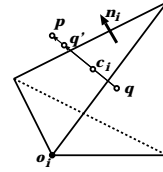


図1 四面体拘束

Fig. 1 Constraint by tetrahedron.

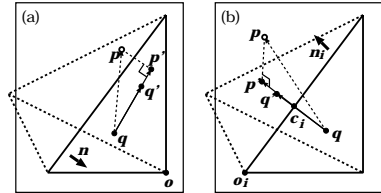


図2 面拘束

Fig. 2 Constraint by surface.

外向き法線を n とし、次式により判定を行う。

$$n \cdot (p - q) > 0 \quad (5)$$

外向きならば、外向き方向に接する四面体の材料 id_{out} を参照し、 $\text{move}(id_{out}, p', q, q')$ により移動できるか調べる。ここに、3自由度の移動が許されることから p' は p に等しい。移動できれば、外向き方向に接する四面体拘束状態に遷移する (q は不変)。

- (a-2) 内向きならば、現在 q が属している四面体の材料 id を参照し、 $\text{move}(id, p', q, q')$ により移動できるか調べる。移動できれば、内向き方向に接する四面体拘束状態に遷移する (q は不変)。

(b) 現在の拘束状態での移動

- (b-1) $\text{move}(id, p', q, q')$ により q が移動するか調べる。ここに、 p' は面拘束状態での目標の指先であり、 p から現在属する面に下ろした垂線の足として定義する。これは、面上の1点を o 、面の法線を n とすると次式により計算される。

$$p' = (p - o) - ((p - o) \cdot n)n + o \quad (6)$$

移動しなければ、現在の拘束状態を維持する (q は不変)。

- (b-2) q' が三角形の範囲内であれば、現在の拘束状態を維持する (図2(a))。現在属する面以外の3面のそれぞれ (i) について面上の1点を o_i 、面の法線を n_i とし、式(2)より外部の判定を行う。すべての面について内部と判定された場合には、現在の拘束状態を維持する ($q = q'$)。

- (b-3) q' が三角形の範囲外であれば、自由度を縮小する (図2(b))。現在属する四面体の4面のうち、現在の拘束面を除くそれぞれ (i) について、面上の1点を o_i 、面の法線を n_i とし、面 i と線分 (q, q') との

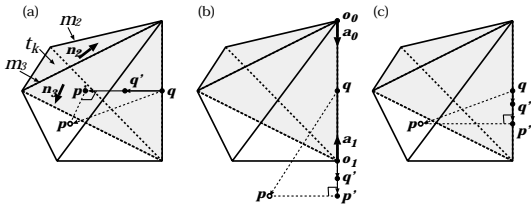


図3 辺拘束

Fig. 3 Constraint by edge.

交点 c_i を式 (3) より求める．交点が見つかった場合には、距離 $|q - c_i|$ を最小にする c_i を選択し、辺拘束状態に遷移する ($q = c_i$)．

2.3 辺拘束

(a) 自由度拡大

(a-1) p の方向への移動ができるか調べる．現在属する辺に隣接する四面体のそれぞれ (t_k) に対して、辺に隣接する面の法線を n_2, n_3 とする．すべての t_k のうち、次式を満たす四面体が p の方向に位置する四面体である．

$$\begin{cases} (p - q) \cdot n_2 < 0 \\ (p - q) \cdot n_3 < 0 \end{cases} \quad (7)$$

p の方向に四面体があれば、その四面体の材質 id_k を参照し、 $move(id_k, p, q, q')$ により q が移動できるか調べる．移動できれば、 t_k による四面体拘束状態に遷移する (q は不変)．

(a-2) p の方向に四面体が見つかることができない場合は、外部状態へ遷移する ($q = p$)．

(a-3) p の方向へ移動できない場合は、辺を共有するこれ以外の四面体への進入の可能性を調べる (図3(a))．現在属する辺に隣接する四面体のそれぞれ (t_k) に対して、辺に隣接する面 (m_2, m_3) の法線を n_2, n_3 、面上の1点を o とする．

m_2 について、 p から m_2 に下ろした垂線の足 p' を式 (6) より計算する． p' が次式を満たしていれば、面 m_2 は指先が移動できる範囲にあると見なし、 $move(id_k, p', q, q')$ により q が移動できるか調べる．

$$(p' - q) \cdot n_3 < 0 \quad (8)$$

移動できれば、 m_2 を候補に追加する． m_3 についても m_2 と同様に調べる．

すべての t_k の m_2, m_3 の中から得られた候補の中で $|p' - q|$ を最大にする面を選択し、選択された面による面拘束状態へ遷移する (q は不変)．

(b) 現在の拘束状態での移動

(b-1) 辺上を移動するかどうか調べる．まず、辺拘束状態における目標の指先位置 p' を求める．上述のよ

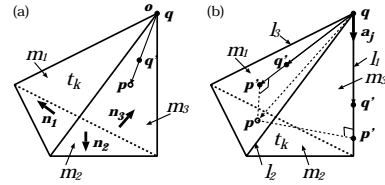


図4 点拘束

Fig. 4 Constraint by node.

うに p' を p から辺に下ろした垂線の足として定義する．これは、現在拘束されている辺上の1点を o 、辺の方向を a とすると、次式により求められる．

$$p' = ((p - o) \cdot a)a + o \quad (9)$$

次に $move(id, p', q, q)$ により q が移動するか調べる．移動しなければ、現在の拘束状態を維持する (q は不変)．

(b-2) q' が線分の範囲外であれば、自由度を縮小する (図3(b))．具体的には、 q' が端点を越えて移動した場合、端点での点拘束状態に遷移する．現在拘束されている辺の1つの端点とその点における線分内向きの方向 o_0, a_0 を用いて次式により外部判定を行う．

$$(q' - o_0) \cdot a_0 \leq 0 \quad (10)$$

上式が成立すれば、 q' は線分の範囲外にあり、 o_0 における点拘束状態に遷移する ($q = o_0$)．同様に他の端点およびその方向 o_1, a_1 について式 (10) により外部判定を行う．式 (10) が成立すれば、 q' は線分の範囲外にあり、 o_1 における点拘束状態に遷移する ($q = o_1$)．

(b-3) q' が線分の範囲内であれば、現在の拘束状態を維持する (図3(c)) ($q = q'$)．

2.4 点拘束

(a) 自由度拡大

(a-1) p の方向への移動ができるか調べる (図4(a))．現在拘束されている点に隣接する四面体のそれぞれ (t_k) に対して、点に隣接する面の法線を $n_1 \sim n_3$ とする．すべての t_k のうち、次式を満たす四面体が p の方向の四面体である．

$$\begin{cases} (p - q) \cdot n_1 < 0 \\ (p - q) \cdot n_2 < 0 \\ (p - q) \cdot n_3 < 0 \end{cases} \quad (11)$$

p の方向に四面体があれば、その四面体の材質 id_k を参照し、 $move(id_k, p, q, q')$ により q が移動するか調べる．移動すれば、 t_k による四面体拘束状態に遷移する (q は不変)．

(a-2) p の方向に四面体が見つけれない場合は、外部状態へ遷移する ($q = p$)．

(a-3) p の方向へ移動できない場合は、頂点を共有

するこれ以外の四面体への進入の可能性を調べる (図 4(b)). 現在拘束されている点に隣接する四面体のそれぞれ (t_k) に対して, 点に隣接する面 m_i ($i \in 1, 2, 3$) の外向き法線を n_i , 面上の 1 点を o , 隣接する辺 l_j ($j \in 1, 2, 3$) の方向を a_j とする. まず面による拘束では, m_1 について, これに拘束された場合の目標の指先 p' を p から m_1 に下ろした垂線の足として式 (6) により計算する. p' が次式を満たしていれば, m_1 は指先が移動できる範囲にあると見なし, $\text{move}(id_k, p', q, q')$ により q が移動するか調べる.

$$\begin{cases} (p' - q) \cdot n_2 < 0 \\ (p' - q) \cdot n_3 < 0 \end{cases} \quad (12)$$

移動すれば t_k の m_1 を候補に追加する. m_2, m_3 についても同様に調べる.

次に辺による拘束では, l_1 について, これに拘束された場合の目標の指先 p' を p から l_1 に下ろした垂線の足として式 (9) により計算する. p' が次式を満たしていれば, l_1 は指先が移動できる範囲にあると見なし, $\text{move}(id_k, p', q, q')$ により q が移動するか調べる.

$$(p' - q) \cdot a_1 < 0 \quad (13)$$

移動すれば t_k の l_1 を候補に追加する. l_2, l_3 についても同様に調べる.

すべての t_k の m_i, l_j の中から得られた候補の中で $|p' - q|$ を最大にする面あるいは辺を選択する. 候補が面であれば面拘束状態に遷移する (q は不変). また, 候補が辺であれば面拘束状態に遷移する (q は不変).

(b) 現在の拘束状態での移動

自由度を拡大する移動を行うことができない場合は, 現在の拘束状態を維持する (q は不変).

2.5 外部

(a) 物体内部への進入

すべての表面三角形について, これと指の軌跡との交差の判定を行う (図 5). 外部では $p' = p$ であり, また無条件に $q' = p'$ である. この中では, 各々の三角形が属する四面体の幾何学的形状を利用する. 各々の三角形についてまず, これを含む平面と線分 (q, q')

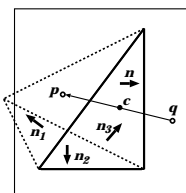


図 5 外部状態

Fig. 5 No constraint (external).

との交点 c を求める. 平面は, 面上の 1 点 o と法線 n により定義され, 計算には式 (3) を用いる. 有効な交点 c が得られた場合, さらにこれが三角形面内にあるか調べる. すなわち, 表面三角形を含む四面体において, 表面三角形以外の 3 面の法線をそれぞれ $n_1 \sim n_3$, 面上の点を $o_1 \sim o_3$ とするとき, 以下の式を評価する.

$$\begin{cases} (c_i - o_1) \cdot n_1 < 0 \\ (c_i - o_2) \cdot n_2 < 0 \\ (c_i - o_3) \cdot n_3 < 0 \end{cases} \quad (14)$$

交点 c に対して上式が成立すれば面拘束状態に遷移する ($q = c$).

(b) 外部状態の維持

物体への進入が検出されなかった場合には, 現在の拘束状態を維持する ($q = p$).

3. 実装と評価

前章で述べた干渉計算のアルゴリズムを仮想空間に実装した. 計算の実時間性を向上するため, あらかじめ計算しておくことのできる幾何学的情報については, これをモデルに保持するものとした. 具体的には, 四面体の各面の外向き法線ベクトル, 各辺の端点における内向き方向ベクトル, 頂点および辺について, これを共有する四面体の集合は干渉計算の実行に先立ち求めておくものとした.

指先の位置姿勢の入力と物体との干渉の際の力覚提示を行うデバイスとして PHANToM (SensAble Technologies) を用いた. 干渉計算および結果の描画は DOS/V PC (Pentium Pro 200 MHz) により行われる. なお, 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 (Fire GL 1000, Diamond Multimedia) を使用した. また, OS としては WindowsNT を用いた. また, デバイスの制御に PHANToM Basic I/O ライブラリを利用した. このライブラリは, 力覚計算部分を視覚計算部分と別のスレッドに分け, これを 1 [kHz] で実行する機能を提供する.

3.1 計算時間の評価

指先の状態遷移にかかる時間の計測を行った結果を表 2 に示す. これは, 指先 1 点がそれぞれの状態遷移に要する時間の実測値である. 辺および点拘束状態からの遷移は, これらを共有する四面体の数によって計算時間が変化する. ここでは, 辺拘束状態ではこの辺に 12 個の四面体, また点拘束状態ではこの点に 20 個の四面体が隣接している場合を想定した. 一方, 外部からの遷移についても表面を構成するすべての三角

表 2 計算時間
Table 2 Computation time in state transition.

現在の状態	遷移後の状態								
	現在属する四面体				隣接する四面体				外部
	四面体	面	辺	点	四面体	面	辺	点	外
四面体	10.6	46.6	×	×	×	×	×	×	×
面	4.8	13.2	36.3	×	4.8	×	×	×	×
辺	6.1	158.3	160.8	163.1	6.1	158.3	×	×	×
点	12.0	489.3	489.3	485.6	12.0	489.3	489.3	×	×
外	×	50.7	×	×	×	×	×	×	50.7

(単位はすべて μsec)

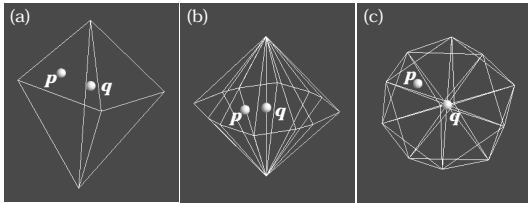


図 6 計算時間測定のためのモデル
Fig. 6 Examples of constraints.

形ポリゴンと交差判定を行うことから、そのポリゴン数に比例する。計算時間の計測では、表面ポリゴン数として 12 を用いた。これは外部領域との境界を立方体により表現した場合の表面ポリゴンの数に等しい。これらの評価条件の例を図 6 に示す。図 6 (a) は四面体が 2 個隣接しており、その境界面に理想の指先が拘束されている様子である。図 6 (b) は上述のように 1 つの辺が四面体 12 個により共有されており、この辺上に理想の指先が拘束されている様子である。図 6 (c) は 20 面体の中心と表面の頂点とをつなぐことにより作成したモデルであり、その中心点に理想の指先が拘束されている様子である。表より、辺拘束から面、辺、点拘束への遷移および点拘束から面、辺、点拘束への遷移で比較的多くの計算時間を必要としていることが分かる。これは、前章でも述べたように、辺または点から移動する際に、これを共有するすべての四面体を対象として移動可能な方向を探索するためである。

3.2 干渉にともなう力の提示

提案した干渉計算手法を用いて干渉にともなう力の提示を行う。まず、指先位置と理想の指先位置との関係から、指先に作用する力を定義する。ここでは、次式に示すように単純に両者の間の距離に比例するものとした。

$$f = k(q - p) \tag{15}$$

ここに、 k は比例定数で、実験では $0.1 [N/mm]$ とした。

次に、材料、目標の指先、現在の理想の指先から移動後の理想の指先を求める移動関数を定義する。ここ

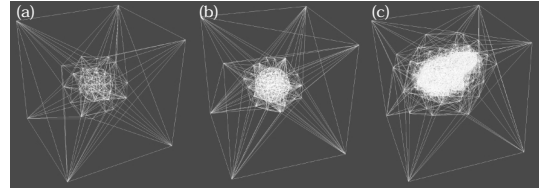


図 7 対象とするモデル
Fig. 7 Tetrahedral elements in the model.

では、以下の 2 つの関数を定義した。

移動関数 (1)

$$q' = \begin{cases} p' & (\text{返り値 } 1, \text{ 材料 } 0) \\ q & (|p' - q| < D) \\ & (\text{返り値 } 0, \text{ 材料 } 1) \\ p' + \frac{D}{|q-p'|}(q - p') & (|p' - q| \geq D) \\ & (\text{返り値 } 1, \text{ 材料 } 1) \end{cases} \tag{16}$$

移動関数 (2)

$$q' = \begin{cases} p' & (\text{返り値 } 1, \text{ 材料 } 0) \\ q & (\text{返り値 } 0, \text{ 材料 } 1) \end{cases} \tag{17}$$

これらの関数は、材料 0 として摩擦力等の発生しない領域を仮定している。また、材料 1 としては、移動関数 (1) では進入することはできるが移動にともなう摩擦力 F を発生するもの¹¹⁾、移動関数 (2) では進入することのできないものを仮定している。実験では $D = 1 [mm]$ (したがって、最大摩擦力 $0.1 [N]$) としている。

対象物体のモデルとしては、(a) 球形状の 80 面体、(b) 球形状の 320 面体、(c) ねずみの形状の 3 種類を用いた。これらのモデルを図 7 に示す。外部から四面体で構成されるモデルへ入る際の処理時間の観点からは、モデルの中で外部と接するポリゴンの数はできるだけ少ないことが望ましい。実験では、上述の形状の周りにさらに立方体形状の空間を定義し、この領域も四面体の要素に分割した。これにより、モデルの表面は 12 個の三角形ポリゴンで構成される。これら 3 つの対象

表3 対象物体の四面体数・頂点数
Table 3 Models for experiments.

	80面体	320面体	ねずみ
四面体数	394	1917	11831
(材料0:外)	234	637	5519
(材料1:内)	160	1280	6312
頂点数	82	353	2099

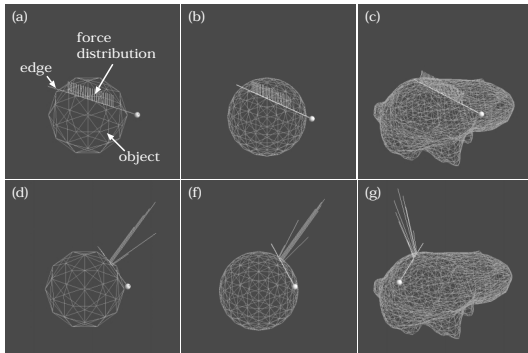


図8 干渉にともなう力の提示

Fig. 8 Representation of constraint force.

物体の四面体数, 頂点数を表3に示す. 80面体, 320面体はそれぞれ正20面体を再分割することにより作成した. ねずみは3次元形状スキャナ(Cyberware)により形状を取得し, 手作業により四面体に分割した. また, 対象物体と立方体との間の空間も手作業により四面体に分割した. なお, 球の半径は30[mm], ねずみの体長はおよそ120[mm]である.

以上のモデルによる力の提示の様子を図8に示す. ここでは, 対象と指先との干渉の状態が分かりやすいように, 複数の指先を直線上に並べ, これら各々と対象との干渉および各々に作用する力の計算を行っている. これはまた, 離散化された刃先と物体との干渉の計算の単純な場合という意味合いを持っている. この図においては, 物体がワイヤフレームで描かれ, 指先の列が直線で示されている. また, 直線付近に描かれている直線群が提示力である. なお, この図では, 対象物体の外部のメッシュは描画を省略している. また, デバイスの制約により操作者には指先と物体の干渉により計算された力の総和のみがフィードバックされ, トルクは提示されない. 図8(a)~(c)は移動関数(1)を用いることで指先に抗力を表現した場合, (d)~(f)は移動関数(2)を用いることで対象物表面での拘束力を表現した場合である. 図において, 直線の長さは0.1[m], 指先数は50である. 図8(d)において, 提示されている力の方向にばらつきがあるのは, 物体表面のポリゴンが滑らかでないために起こる現象である.

図8(e)の320面体は(d)の80面体よりも表面ポリゴンの相互になす角度が小さいため, 提示力の方向のばらつきが少ないことが分かる.

計算時間の評価の中で明らかにしたように, 辺, 点からの状態遷移には多くの計算時間を要する. しかしながら, 50点の指先について処理を行っている上の例においても, およそ1[kHz]の更新レートでの干渉計算が可能であった. これは, 辺, 点での拘束は一般には他の拘束状態に比べて起こる確率が低いいため, 力の計算の実時間性に影響が少なかったためであると考えられる.

4. おわりに

本研究では, 実時間性と, 力学的規則に基づく指先移動およびその連続性を考慮した指先と空間との干渉計算方法について検討した. 空間を四面体要素の集合として表現し, この表現に基づいて干渉計算を行う手法を提案した. また, 指先の移動にともない指先が属している要素を検索し, 物体の材質に基づいて指先を移動するアルゴリズムを示した. これらを仮想空間に実装することで, 実時間での指先と物体との干渉計算が可能であることを確認した. また, 指先の移動に力学的条件をつけることにより摩擦力を表現し, 四面体に進入できる材質とできない材質を与えることにより固い物体の上を滑るという現象を実現した. さらに, 物体だけでなく周りの空間も四面体で表現することにより, 外部からの遷移の高速化を実現した. 実験で用いたモデルでは, 物体の周りに立方体領域を定義していた. これは, 外部との境界をなす表面ポリゴンを少なくすることで外部からの遷移を高速化することを目的としていた. この意味からは, 表面を四面体とすることでこれを4ポリゴンで定義することができ, したがってこの遷移の処理時間がさらに $\frac{1}{3}$ 程度となることが期待できる.

今後の課題としては, 辺または点拘束状態からの遷移計算の高速化をあげることができる. これらの遷移に比較的多くの計算時間を必要とするのは, 前述のように辺または点から移動する際に, これを共有するすべての四面体を対象として移動可能な方向を探索するためである. この問題を解決する1つの方法として, 辺および点に隣接する四面体数をできる限り少なくすることが考えられる. このためには, 空間を四面体へ分割する際の隣接四面体数に基づく最適化手法の検討が必要である.

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) Burdea, G.: *Force & Touch Feedback for Virtual Reality*, A Wiley-Inter-Science Publication (1996).
- 3) Massie, T.H.: Initial Haptic Explorations with the Phantom: Virtual Touch Through Point Interaction, Master's thesis, M.I.T. (1996).
- 4) Shimoga, K.: A study of perceptual feedback issues in dextrous tele-manipulation: Part I. Finger force feedback, *VRAIS'93*, pp.263-270 (1993).
- 5) Kitamura, Y., Smith, A., Takemura, H. and Kishino, F.: A Real-Time Algorithm for Accurate Collision Detection for Deformable Polyhedral Objects, *PRESENCE*, Vol.7, No.1, pp.36-52 (1998).
- 6) Zilles, C.B. and Salisbury, J.K.: A Constraint-based God-object Method For Haptic Display, *IROS'95*, pp.145-151 (1995).
- 7) Ho, C., Basdogan, C. and Srinivasan, M.A.: Haptic Rendering: Point- and Ray-Based Interactions, *PUG'97* (1997).
- 8) 北川英志, 安田孝美, 横井茂樹, 鳥脇純一郎: 仮想空間操作を利用した対話型手術シミュレーションシステムの基本機能の実現, 情報処理学会論文誌, Vol.37, No.6, pp.1088-1098 (1996).
- 9) Galyean, T.A.: Sculpting: An Interactive Volumetric Modeling Technique, *SIGGRAPH '91*, Las Vegas, pp.267-274 (1991).
- 10) Yamamoto, K., Ishiguro, A. and Uchikawa, Y.: A Development of Dynamic Deforming Algorithms for 3D Shape Modeling with Generation of Interactive Force Sensation, *VRAIS'93*, Seattle, U.S.A., pp.505-511 (1993).
- 11) 田中厚子, 広田光一, 金子豊久: 仮想切断作業における力覚の表現, 情報処理学会論文誌, Vol.40, No.7, pp.2978-2987 (1999).

(平成 11 年 3 月 8 日受付)

(平成 11 年 11 月 4 日採録)



田中 厚子

1997 年豊橋技術科学大学工学部情報工学科卒業。1997 年同大学院修士課程修了。同年オムロン(株)に入社、現在、技術本部中央研究所イメージラボに勤務。在学中は仮想空間を利用した対象物操作の研究を行う。現職では 3D 画像を使った車両認識の研究開発に従事。日本 VR 学会会員。



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。