

# 触覚ディスプレイのための基本ソフトウェアの開発

## The development of Haptic Interface Platform

廣瀬 通孝<sup>1)</sup> 岩田 洋夫<sup>2)</sup> 池井 寧<sup>3)</sup> 小木 哲朗<sup>4)</sup> 広田 光一<sup>5)</sup> 菊池 望<sup>6)</sup>

Michitaka HIROSE Hiroo IWATA Yasushi IKEI Tetsurou OGI Kouichi HIROTA Nozomu KIKUCHI

- 1) 東京大学工学部総合試験所 (〒113 東京都文京区本郷七丁目三番一号 E-mail:hirose@ihl.t.u-tokyo.ac.jp)
- 2) 筑波大学構造工学系 (〒305 茨城県つくば市天王台 1-1-1 E-mail:iwata@kz.tsukuba.ac.jp)
- 3) 東京都立科学技術大学工学部生産情報システム工学科  
(〒191 東京都日野市旭が丘 6-6 E-mail:ikei@krmgiks5.tmit.ac.jp)
- 4) 東京大学インテリジェントモデリングラボラトリ  
(〒113 東京都文京区弥生 2-11-16 E-mail:tetsu@iml.u-tokyo.ac.jp)
- 5) 豊橋技術科学大学情報工学系  
(〒441 愛知県豊橋市天伯町雲雀ヶ丘 1-1 E-mail:hirota@axia.tutics.tut.ac.jp)
- 6) 日商エレクトロニクス株式会社ビジュアルコミュニケーション事業部画像電子部  
(〒104 東京都中央区築地 7-3-1 E-mail:kikuchi@nissho-ele.co.jp)

ABSTRACT. Recently, several haptic devices have been commercially available. However, a software for these haptic devices is not sufficiently prepared. In this project, Haptic Interface Platform(HIP) that is a software library independent on the types of the haptic devices were developed. It provides several haptic functions based on surface model, texture model, dynamics model, deformation model, and spatially distributed model.

### 1. 背景

人工現実感技術の進展に伴って、仮想の触覚感を作り出す触覚ディスプレイが利用できるようになってきた。しかしながら、これらのディスプレイを使って、何が表現できるか、何を表現したら面白いのかというソフトウェアを中心とした議論はまだ開始されたばかりの段階である。今後、たとえば電子博物館を始めとする電子的なアーカイビングを行おうとしたとき、触覚情報を一般化された表現で保存したり、また遠方に伝達して再生できるように基本ソフトウェアが必要となる。

本プロジェクトでは、このような観点から触覚情報を表現するための基本ソフトウェア HIP(Haptic Interface Platform)の開発を行った。以下、本稿では、HIPの構成及び各機能の表現方法などについて説明する。

### 2. 目的

HIP は、計算機によって触覚を表現するためのソフトウェアを体系化し、様々な触覚ディスプレイの形式で触覚情報を提示できるように設計された汎用ライブラリである。従来の視聴覚用の提示ソフトウェアと合わせてHIPを利用することにより、仮想空間のコンテンツを見たり聞いたりするだけでなく、触ることもできるように提示することを目的とする。

### 3. HIPの構成及び各機能の表現方法

HIPの全体構成を図1に示す。本章では、HIPの各機能の表現方法について詳述する。

#### (1) ポリゴンモデルを用いた触覚表現

最も簡単な触覚表現として、仮想物体と指先とのイン

タラクションによる反力の表現があげられる。これは、指先位置と仮想物体との接触判定計算を行い、その結果、物体形状を表現するに適切な反力を計算することである。このためには物体の表面形状を記述する触覚モデルが必要になるが、HIPでは物体形状を、ポリゴンにより記述する汎用フォーマットであるVRMLファイルを読み込み、触覚モデルとした。

#### a) 表面接触点

ポリゴンモデルを触る触覚レンダリング手法としては"God-Object Method"を導入した[1]。これは、指先の動きに応じてポリゴンモデル表面上を滑るように動く仮想の点である表面接触点を設定する方法である。この表面接触点の位置を正確にコントロールすることで、反力の方向を決定することが容易になる。

以下に表面接触点を用いた触覚レンダリングの手順を示す。

#### ・指先とポリゴンモデルとの接触面の決定

まず、表面接触点がどここの面上にあるかを決定することが必要になる。指先がはじめて面に触れる場合は、指先位置の履歴をとることで、接触面は特定できる。次に1面に接触した後、指先を動かしていくと、接触面は面のつなぎ目で移り変わる。図2はポリゴン間が凸な場合と凹な場合について隣接面の特定方法を示したものである。

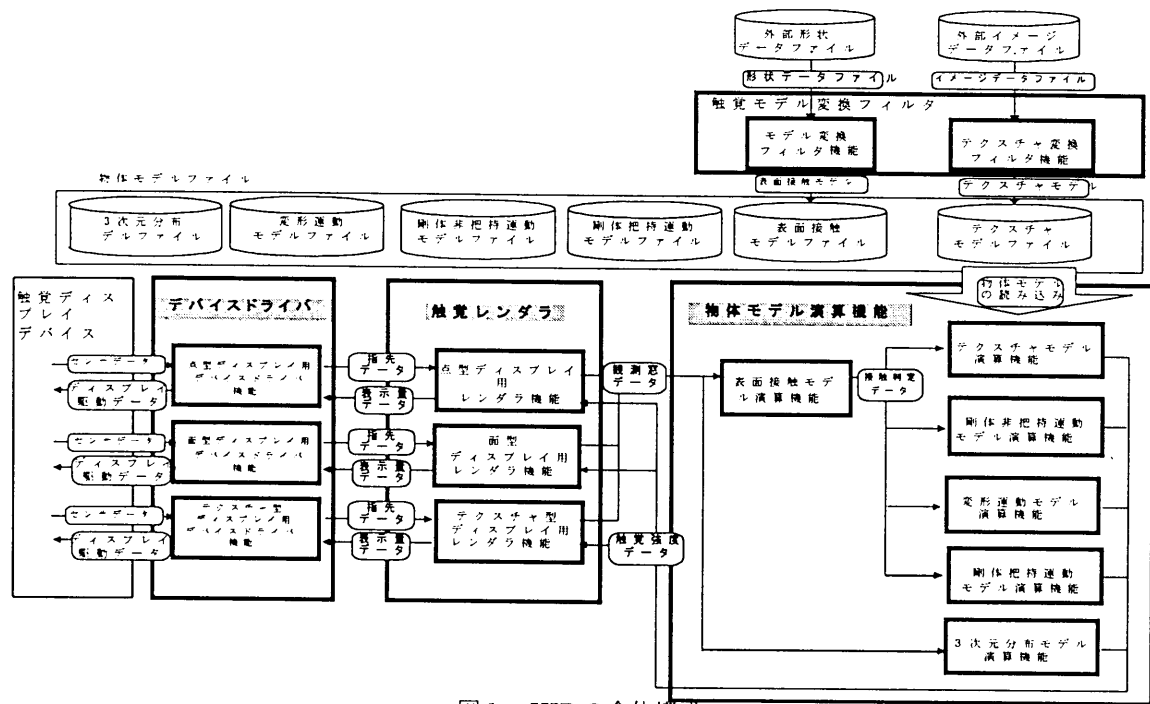


図1 HIPの全体構成

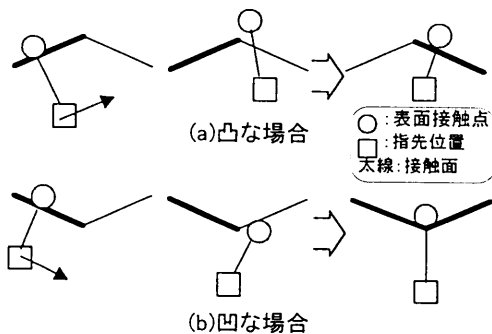


図2 表面接触点の接触面推移

・表面接触点の位置更新

指先位置と表面接触点との間の距離が最小になるようにするために、表面接触点の位置を更新する。更新計算は、Lagrange 乗数法を用いて計算する。指先位置を  $(x_p, y_p, z_p)$  とし、表面接触点の位置を  $(x, y, z)$  とすると、この間のバネエネルギー  $Q$  は次式のように定義され、この値が最小になるように表面接触点の位置が計算される。

$$Q = \frac{1}{2}(x - x_p)^2 + \frac{1}{2}(y - y_p)^2 + \frac{1}{2}(z - z_p)^2 \quad (1)$$

・反力の計算

表面接触点の位置と指先位置が決まれば、フックの法則に従い、反力を以下の式により計算する。

$$(\text{反力}) = (\text{面の弾性係数}) * ((\text{表面接触点の位置}) - (\text{指先位置}))$$

b) 接触判定計算の高速化

触覚提示には、非常に速い制御速度が要求される。上記で紹介したアルゴリズムは、接触時には、接触面に隣接した面との接触判定を行うだけであるが、非接触時には、すべての面との接触判定計算を行うため、非常に効率が悪い。

そこで、HIP ではオクトツリーを用いた接触判定の高速化を行った。オクトツリーとは、モデルのバウンディングボックスを再帰的に8分割していき、細分化された

ボックスに含まれる面数が、ある個数になったら分割を終える。そのため、図3に示すように、面が密な箇所ほど細かく、疎なほど粗く分割される。分割されたボックスの中に含まれる面番号は、前処理の段階で記憶される。

接触判定時には、指先がどのボックスに含まれるかを検索していき、指先が含まれるボックスの中の面とだけ接触判定を行うことによって、高速化できる。

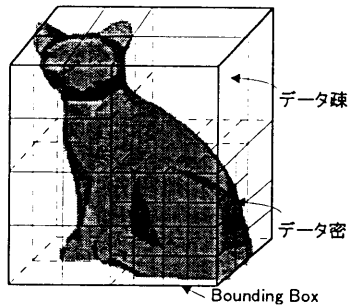


図3 オクトツリーの概念

c) フォースシェーディング

曲面を触覚で表現する場合、ポリゴンモデルで記述すると、どうしてもつなぎ目で反力の方向が変化してしまい滑らかに感じられない。そこで、CG のグローシェーディングと同じような手法による反力のシェーディングを実装した。これは、各頂点の法線ベクトルを計算し、ポリゴンの表面接触点での法線ベクトルを線形補間するという方法である。図4はシェーディングの方法を图示したものであり、提示力は次式で表せる。

$$f = \Sigma(x(\Sigma n_i - n_i)) / \Sigma x_i \quad (2)$$

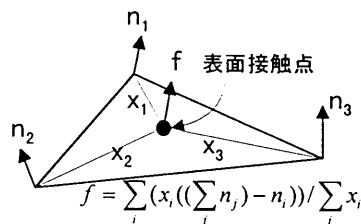


図4 フォースシェーディング

## (2) テクスチャ情報の表現

テクスチャ型のディスプレイを用いる場合は、対象の表面の微細な凹凸などに起因するテクスチャ（肌理）の感覚を提示することが可能である。

### a) テクスチャ表現の流れ

テクスチャ表現の計算は、「テクスチャモデル演算機能」と「テクスチャ型触覚レンダラ」によって行われる。一般の触覚モデル演算では、入力に対する対象物体のレスポンスを計算するという見方が分かりやすいが、テクスチャの場合は、対象の内部状態の変化はなく、面内のピン位置に対応する触覚強度の単純な読み出しとみなすことができる。テクスチャモデルの演算に先立って、表面接触モデルの計算が行われ、対象物体との接触状態のデータがテクスチャモデルに与えられる。レンダラは、指先の位置情報をモデルに与え、その提示量をモデルから切り出す機能を持ち、これは仮想世界におけるインタラクションの情報を指モデル間で動かす部分に相当する。

### b) テクスチャモデル

テクスチャモデルは、対象から受ける触覚的的刺激量を矩形領域のマップとして保持する。このマップは、テクスチャディスプレイで表示したときに妥当な表現となるように予め調整されたものである。モデルの表現においては、離散型と連続型の2つの形式が用いられる。離散型とは、画像のテクスチャと同様に、ピクセルに相当する触覚要素の集まりで領域を記述する形式であり、連続型とは、領域を2次元の座標でインデックスし、その座標の関数として触覚強度を記述する形式である。

### c) テクスチャの提示例

HIP を用いて触覚テクスチャの提示を行った例を図5に示す。

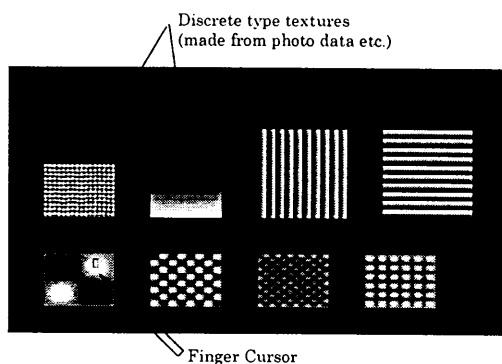


図5 テクスチャの提示例

描画テクスチャの輝度値が触覚強度に対応している。なお、ここでは、机上の水平面が、視覚ディスプレイに平行な面として描画されている。左上の2つが、離散型のテクスチャモデルであり、その他は、関数によって記述された連続型テクスチャモデルである。左端の離散型テクスチャモデルは、凹凸のある実物の物体表面をデジタルカメラで撮影したデータに基づくものであり、その右のモデルはグラデーション画像から作成されたものである。このような画像データをもとにテクスチャモデルを作成するためのイメージファイル変換フィルタが、HIP のツールとして用意されている。

下段のテクスチャは、三角関数による滑らかな凹凸の刺激分布であり、上段右の大き目のテクスチャは、鋸波

状である。連続型は、C の関数として記述できるものであれば良いので、例えば、時間変動するランダムな表面形状なども提示することができる。テクスチャディスプレイの移動量に応じて動く指先カーソルの部分の触覚強度が、モデルからサンプルされて、振動ピンの振動強度に反映される。

## (3) 剛体運動の表現

「触覚表現」の中で、単純であるが用途の広いものとして、対象物を剛体とみなした場合のインタラクションの実現があげられる。これは、変形しない硬い物体を押して動かしたり、これを掴んで振り回したりする場合の力の感覚の表現を意味する。

### a) 剛体運動モデル

剛体運動モデルでは基本的には、剛体の運動特性の定義と、これを用いた運動方程式の計算とが行われる。HIP ではこのような運動を、操作者が対象を掴んでいる場合とそうでない場合とに分けて考えるものとした。

これら2つの状態のシミュレーション計算における違いは、力から運動を計算するか、運動から力を計算するかという点にある。以下では、各々の場合について計算方法を説明する。

### b) 非把持状態の運動計算

非把持状態では、指先などと対象物との干渉にともなって斥力が発生し、これが操作者に操作力として知覚されると同時に、これにもとづいて対象物の運動状態が変化するという考え方を採る。力から運動を計算することから、これは順動力学計算となる。

#### ・非把持状態作用力演算

力提示-位置計測型のデバイスは、操作者と仮想物体との間に作用している力などの情報を直接計測することができない。作用力演算の処理では、物体表面に仮想の弾性を仮定し、指先が物体表面からめり込んでいる距離にこの弾性定数を乗じることで、作用力を計算する。

#### ・非把持状態触覚強度演算

HIP における触覚強度とは、操作者に帰還される触覚情報情報の総称であり、剛体運動モデルにおいては、指先位置ごとの力およびトルクをさす。これは、作用力演算で求められた力の反作用として計算される。

#### ・非把持状態対象物運動演算

慣性行列に代表される対象物の運動パラメータは重心座標で定義され、これらの計算も重心座標系を用いて行われる。

はじめに、作用力演算で求められた力をもとに、重心座標系における原点回りの力およびトルクが計算される。また、重力加速度が重心座標系での値に変換される。つぎに、ニュートンおよびオイラーの運動方程式をシミュレーションのサイクルタイム（離散時間）で離散化し、これに力およびトルクを代入することで、物体の加速度および角加速度を求める。さらに、この加速度および角加速度をもとに、離散時間後の速度・角速度および位置変化・角度変化を求める。最後に、位置変化・角度変化をもとに物体の位置・姿勢を更新する。

### c) 把持状態の運動の計算

把持状態では、指先など対象を把持している点の移動が直接に対象物の運動状態に反映されるものと考え、この運動状態の変化を生じさせるような力およびトルクが指先などから与えられたと仮定し、さらに、この力およびトルクの反作用力が操作者に知覚されるものとする。運動から力を計算することから、これは逆動力学計算となる。

#### ・把持状態対象物運動演算

対象物運動演算では、物体の姿勢の時間変化を求める必要があるが、これに先立ち、指先の位置・姿勢と物体の位置・姿勢との関係を定義する必要がある。

物体の位置・姿勢と1サイクル前の位置・姿勢とから、差分計算により運動の速度・角速度を計算し、これと現在の位置・姿勢とを合わせて、現在の運動状態とする。さらに、現在の運動状態と過去の運動状態から差分計算により物体の加速度・角加速度を計算する。

物体の位置・姿勢が仮想世界の座標系(世界座標)と物体に固定された座標系(物体座標系)とのあいだの座標変換として定義されることから、以上の計算は世界座標系で行っているが、次に述べる作用力演算では重心座標での値が必要となることから、最後に座標変換によりこれを求めている。

・把持状態作用力演算

対象物運動演算で求められる加速度・角加速度などの値を運動方程式に代入することで、この物体に作用している力とトルクの総和を計算する。また、つぎに述べる触覚強度演算を物体座標で行っていることから、これらの値を物体座標系での記述に変換する。

・把持状態触覚強度演算

触覚強度演算では、作用力演算でもとめた力およびトルクを指先位置における力およびトルクに分配する。この計算は、物体を把持している指先の数によって、計算が異なってくると考えられる。すなわち、1指の場合はこの点を介してすべての力およびトルクが与えられたと仮定できるが、2指以上では指先の力およびトルクの自由度が分配にともなう制約の自由度を上回るため、解が一意に決定できなくなる。したがって、何らかの仮定をおいた計算が必要となる。HIP では指先が2点の場合には、これらの点に力と、この2点を通る軸の周りにモーメントとが作用したと考える。また、指先が3点以上の場合には、指先の定義されている順序で3点までを選び、その各点に力のみが作用したものと考える。最後に、これらの仮定にもとづいて得られた力およびトルクをユーザの指先が記述される世界座標に変換する。

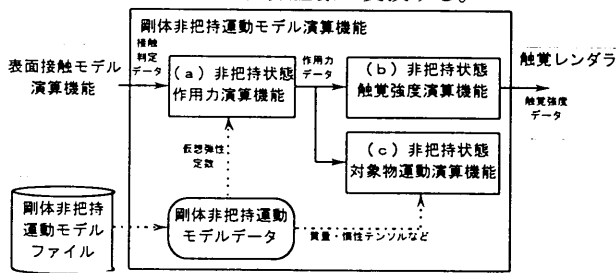


図6 (a)非把持状態の運動の計算

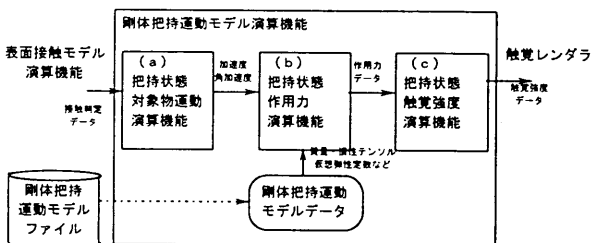


図6 (b)把持状態の運動の計算

(4) 物体の変形運動の表現

ここでは、触覚シミュレーションにおける変形運動の表現方法について説明する。

a) 物体の変形運動

一般に、物体の変形運動は条件によって弾性変形、塑性変形等の複雑な現象を示す。

変形運動のシミュレーションでは、状況に応じてこれらの現象を精度よくシミュレーションできることが必要であるが、同時に計算をリアルタイムで実行できることも重要である。HIP では、触覚による変形運動を表現するためのモデルとしてバネモデルと有限要素モデルを導入した。

b) バネモデル

バネモデルは、簡略なモデルにより計算のリアルタイム性を優先した方法である[2]。この方法では、図7に示すように物体の表面を格子状に分割し、各格子点間をバネとダンパで結合したモデルで定義する。物体に力が作用し、ある格子点が移動すると、周囲の格子点もバネによって引っ張られ、全体としてなめらかな変形を実現することができる。

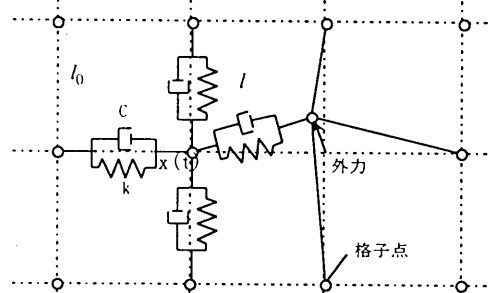


図7 物体表面に定義されたバネモデル

このとき、各格子点の移動は以下の数式によって計算される。

$$f(t)=F(t)+Sk(l-l_0) \tag{3}$$

$$a(t)=(f-cv(t-1))/m \tag{4}$$

$$v(t)=a(t)\Delta t-v(t-1) \tag{5}$$

$$x(t)=v(t)\Delta t+x(t-1) \tag{6}$$

ここで  $F$  は格子点に働く外力で、物体に対する指先のめり込み量によって決定される。 $k$  はバネ定数、 $l$  はバネの長さ、 $l_0$  はバネの自然長で、 $f$  はバネの復元力を加えた格子点に働く力を表している。また  $m$  は格子点の質量、 $c$  はダンパの減衰係数、 $a$ 、 $v$ 、 $x$  はそれぞれ格子点の加速度、速度、変位を表している。式(3)によって格子点に働く力が求められると、これを運動方程式にあてはめ、式(4)、式(5)に従い加速度、速度が求められる。さらに式(6)により現在位置に加算することで、次の瞬間の格子点の位置が求められる。この計算を時間ステップ  $\Delta t$  ごとに繰り返すことで、全体として自然な変形をシミュレーションすることができる。

この方法は物体の材料特性等を考慮したものではなく、現実世界の現象を忠実に再現するものではないが、バネ定数、減衰係数、質量等のパラメータ値を調整することで、いろいろな変形の様子を実現することが可能である。

c) 有限要素モデル

有限要素モデルは、有限要素法による数値解析を触覚シミュレーションの中に組み込んだ方法であり、前述のバネモデルに比べて正確な変形運動を表現可能なモデルである[3]。有限要素法では、連続体である構造物を多数のメッシュに分割し、有限要素の集合体として解析する。解析の基礎となる方程式は、

$$f=Ku \tag{7}$$

で表されるマトリックス方程式である。ここで、 $f$  は外力ベクトル、 $K$  は剛性マトリックス、 $u$  は変位ベクトルを表している。外力が与えられると、式(7)を連立方程式として解くことにより、各節点の変位、すなわち変形形状を求めることができる。

有限要素法では、問題に応じて種々の要素を使用した方法が存在するが、HIP ではこのうち2次元線形問題に対するソルバーを提供している。しかしながら、有限要素法解析の核となる部分は別プロセスとして独立しているため、プロセス間のインタフェースを合わせることで、他のソルバーを組み込むことも可能である。また、プロセスを分けているため実行する計算機を分散させることが可能であり、スーパーコンピュータ等の高速な計算機が使用可能な場合には、有限要素法の計算部分だけを別計算機で実行することもできる。図8は例として、梁の変形計算に対する有限要素モデルを示している。

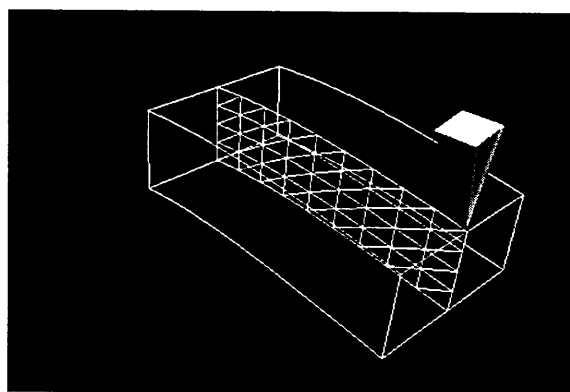


図8 梁の変形計算に対する有限要素モデル

#### (4) 空間分布データの表現

ここでは、流体の解析結果などの空間分布データを触覚表現するために必要な基本的な機能について紹介する。

##### a) 空間分布データ

空間分布データは、任意の位置の状態量をベクトルやスカラー量であらわしたもので空間に分布して定義されているデータであり、物体周りの流体の流れ場や、圧力分布、温度分布、CT 画像のような3次元濃度分布などがある。

空間分布データを使った触覚呈示の基本的な処理は次のようになる。

- ・空間分布データをメモリー上に展開 (初期設定)
- ・現在の触覚デバイスの位置取得
- ・空間分布データの座標系への位置座標変換
- ・空間分布データの取得
- ・得られた空間分布データより触覚情報を計算し出力

以下、空間分布データの処理に必要な「位置座標を空間分布データ座標系に変換する機能」と「その位置での空間分布データを検索する機能」の2つについてそれぞれ詳述する。

##### b) 空間分布データ座標への座標変換

一般に仮想空間の座標系と空間分布データの座標系は、原点やスケールが異なることや、 $x, y, z$  の直交座標系ではなくねじれた座標系になっている場合や各軸のデータの数が異なる場合が多い。このような座標系のデータを扱うために、与えられた  $x, y, z$  軸座標の位置データから空間分布データ空間での座標値を計算する機能が必要である。触覚呈示では情報のアップデートレートが視覚系に比べて数倍から数百倍早くなくてはならないため、なるべくデータ構造は立方格子上の座標点で定義することが望ましい。このようなデータでは、与えられた座標値を  $(x, y, z)$  、データの領域を  $(\min_x, \min_y, \min_z)$  から  $(\max_x, \max_y, \max_z)$  、 $x, y, z$  軸方向のデータの数を  $\text{num}_x, \text{num}_y, \text{num}_z$  とすると、 $(x, y, z)$  からデータ空間の座標値  $(X, Y, Z)$  への座標変換は、以下の式で行うことが出来

る。

$$X = (x - \min_x) / (\max_x - \min_x) * \text{num}_x \quad (8)$$

$$Y = (y - \min_y) / (\max_y - \min_y) * \text{num}_y \quad (9)$$

$$Z = (z - \min_z) / (\max_z - \min_z) * \text{num}_z \quad (10)$$

##### c) 空間分布データの検索と補間

空間分布データの構造には、大きく分けて2つの方式が考えられる。空間に離散的にデータを定義しておく離散型空間分布データと、現在の位置データから空間分布データを関数などを用いて計算する関数型空間分布データの2つである。

##### ・離散型空間分布データ

離散型空間分布データを扱う場合は、空間をある程度離散化させてデータを持たせ、データの定義されていない場所については近傍のデータから補間する方法で空間分布データを呈示する。 $(X, Y, Z)$  のデータを近傍の格子点のデータを用いて補間する方法としては、例えば次の体積補間の式が考えられる。

$(X, Y, Z)$  の空間分布データ =  $\sum$  ((頂点  $P_i$  での空間分布データ) \* (頂点  $P_i$  の対角の頂点が含まれる直方体の体積比))

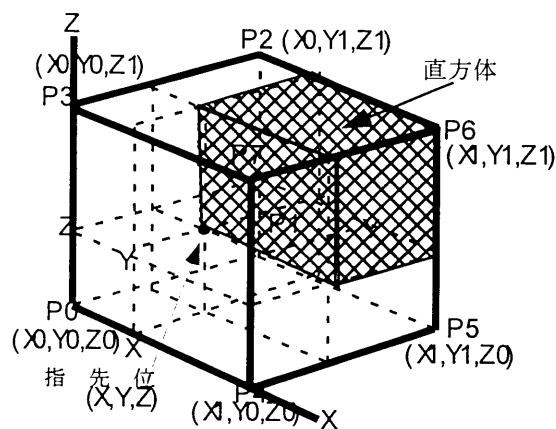


図10 立体格子データの中の指先位置が含まれる単位立方体

##### ・関数型空間分布データ

現在の位置情報から関数などを用いて計算によって求めることが出来るデータを、関数型空間分布データと呼ぶ。現在の指先の座標データを入力するとその位置での空間分布データを出力する関数を定義することが出来れば、どのような空間であっても定義することが可能となる。関数型空間分布データは、離散型空間分布データに比べて演算にかかる時間が増えるが、離散型が補間をしているのに対して、位置データから直接空間分布データの値を計算するので精度が良くなるというメリットがある。

#### 4. おわりに

HIPは、さまざまな触覚的コンテンツを効率的に提示することを目標に開発されたライブラリである。現在利用できる幾つかの触覚ディスプレイによって、上に挙げた6種類の物体モデルを提示することが可能となっている。今後は、物体モデルとディスプレイタイプの独立性を高める拡張や、新しい触覚ディスプレイに対するデバイスドライバの作成などを進める予定である。

5. 参加企業及び機関

東京大学

筑波大学

東京都立科学技術大学

豊橋技術科学大学

日商エレクトロニクス株式会社

6. 参考文献

[1] Zilles C.B., Sallisbury J.K,

"A Constraint-based God-object method for haptic display",

[2] 矢野, 岩田 : 自律的自由曲面を用いた仮想環境における協調作業, 電気学会論文誌 C, Vol.115, No.2, pp.245-252, 1995

[3] 小木, 算, 廣瀬 : 有限要素モデルに基づいた弾性体世界の力覚表現, 日本機械学会第 74 期通常総会講演会講演論文集(IV), pp.158-159, 1997