

# 力覚帰還型仮想環境構築ソフトウェア

矢野 博明<sup>\*1</sup>

岩田 洋夫<sup>\*2</sup>

Software Tools for Haptic Virtual Environment

Hiroaki Yano<sup>\*1</sup> and Hiroo Iwata<sup>\*2</sup>

**Abstract** - Force feedback plays an important roll in recognition and manipulation of virtual objects. Recently haptic devices such as PHANToM and HapticMaster are manufactured. In accordance with rapidly growing network technologies, we can easily get electric media resources :picture and movie and VRML and various digital sound data and so on. However, those electric media resources can only be seen and heard. We can't touch them because there is no software to integrate haptic devices and those electric media resources. In this research , we developed a system that enables users to easy construction of virtual environment with force feedback.We propose that the software for haptic device should have following characteristics. (1) haptic models and haptic rendering function. (2) managing engine of virtual object's behavior. (3) user interaction methods. We implemented the software called Virtual Environment Construction System(VECS).By using this system, we developed 3D shape modeling environment.Through userability study by a large numbers of subjects, effectiveness of the haptic environment is examined.

**Keywords** : haptic, force feedback, force display, virtual environment, software tool

## 1. 背 景

力覚は視覚、聴覚と並んで重要な感覚である。ものに触れたときの重さや硬さといった感覚は、ものを認識し、加工したり操作するときに重要な役割を果たしている。人工現実感の分野ではこのような力覚を与えるための装置をフォースディスプレイと呼んでいる。力覚は、視覚や聴覚と違い感覚受容器が体全体に分散しており、それら全てに感覚情報を呈示する事は困難で、未だにフォースディスプレイの決定的な手法は見いだされていない。とはいえ、最近ではSensAble Technologies社のPHANToMや日商エレクトロニクス社のHapticMasterなど、いくつかフォースディスプレイも実用化され商品として売り出されるようになってきた。このようなハードウェア環境の整備に伴い、アプリケーションを開発するためのソフトウェアツールの重要性が増しつつある。

一方、音声、文字、静止画、動画等の視聴覚情報をデジタル化し、計算機によってインタラクティブに扱う電子メディアの台頭が近年著しい。さらにインターネットの急速な普及に乗じて、世界中の電子メディア情報を手軽に手に入れることができるようになってきている。しかしこれらは、見る、聞くということではできるが、触るとい

う事が出来ない。これは、フォースディスプレイが普及していないことと共に、フォースディスプレイと電子メディア情報を組み合わせるソフトウェアの基本的な枠組みが整備されていないことが原因となっている。

このような視座に立つて本論文では、フォースディスプレイを用いた力覚帰還型仮想空間を構築するためのソフトウェアの基本構成について述べる。さらにこれに基づいて実装された仮想環境を多人数の被験者に体験してもらうことにより、その有効性を検証している。

## 2. 従来のソフトウェアツール

仮想環境を構築するためのソフトウェアツールは、従来より様々なものが開発されている。大学等の研究機関で開発されたものには、ワシントン大学のVEOS、東京大学のVisAge[1]、クラスを用いた仮想環境構成法[2]等がある。商用のものとしては、SENSE8社のWorld-ToolKit[3]、Superscape社のVRT[4]、DIVISION社のdVS[5]等が著名である。これらは、仮想世界の記述方式や仮想物体の挙動の管理について様々な手法を提案している。しかし、感覚フィードバックという観点から見ると、これらのソフトウェアは視覚情報を主として生成するもので、フォースディスプレイの接続についての具体的な問題については論ぜられていない。現在のところ力覚フィードバックに対応した例としては、ノースカロライナ大のArmlib[6]や商用のものとしてSensAble Technologies社のGHOST[7]等を挙げる事ができ

\*1: 筑波大学大学院 工学研究科

\*2: 筑波大学 構造工学系

\*1: Graduate School ,Univ. of Tsukuba

\*2: Institute of Engineering Mechanics ,Univ. of Tsukuba

る。Armlib は、Ethernet を用いてフォースディスプレイとワークステーションを接続している。フォースディスプレイにはプローブの近傍の仮想空間の形状データのみを送り、フォースディスプレイの制御と仮想環境を管理するプログラムのループを分離することで、よりきめの細かい力覚の制御を行うシステムである。GHOST は C++ のライブラリとして提供されたツールキットである。両者とも衝突検出や、重力や摩擦力などの物理現象をサポートしている。

著者らの研究室では、力覚フィードバックに対応した仮想環境構築ソフトウェアの開発に 1992 年より着手し、様々なアプリケーションを通じて熟成を重ねてきた。[8], [9], [10], [11], [12], [13], [14]。本研究で開発したシステムと GHOST との違いは、様々な形式のフォースディスプレイが接続可能なことと視覚呈示系とのマッチングが考慮されていることである。GHOST は原理的には、複数形式のフォースディスプレイを接続可能であるが、現段階でその事例はない。Armlib との違いは各種の力覚プリミティブを備え、高度なユーザ環境が実現できることである。さらに GHOST と Armlib いずれとも異なる点として、複数ユーザをサポートし、協調作業を前提とした設計となっていることが挙げられる。

### 3. 力覚ディスプレイ用ソフトウェア構成

従来の力覚ディスプレイ用ソフトウェアはハードウェアに密着していた為、プログラムの再利用が難しかった。そのためアプリケーションの作り替えに多大な労力が必要であった。近年、計算機の処理能力は急速に向上しており、アプリケーションごとにプログラムの最適化をしなくても十分な性能を発揮できるようになってきている。プログラムの再利用を考慮して、いくつかのモジュールに分けてプログラムを作成し、関係のある場所のみを変更することで作業効率が上がる。力覚ディスプレイ用のプログラムのメインループ中の処理は以下のものである。

- (a) デバイスから手先情報（位置姿勢等）を取り出す。
- (b) 仮想物体等との接触判定を行う。
- (c) 何らかの法則に従って仮想物体の状態を変化させたり、デバイスの状態（形状、機能）を変える。あるいはコマンドの入力を行う。
- (d) 仮想物体等から受ける反力を計算する。
- (e) デバイスに反力を呈示させる。

通常のプログラムでは、以上の処理を繰り返す。これらを機能別に分類すると、(b),(d),(a),(e) は、反力を計算するためのモジュール、(b),(c) は、仮想物体の挙動計算モジュールおよびユーザと仮想環境の相互作用を行うためのモジュールとして考えることができる。したがって力覚ディスプレイ用ソフトウェアには、以下の 3 つの

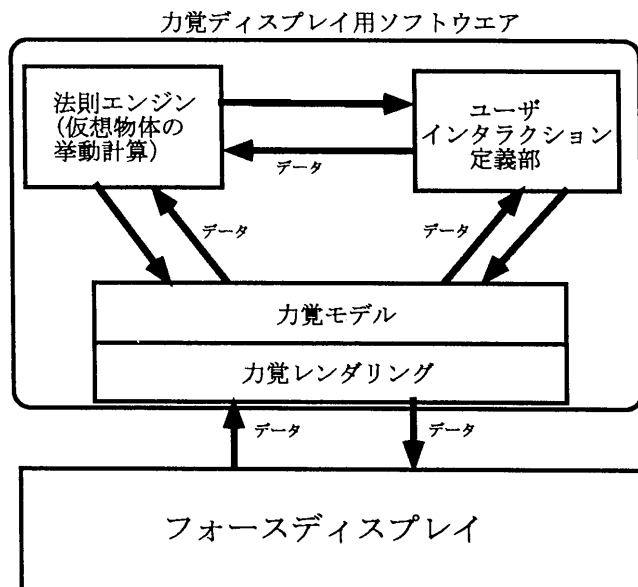


図1 力覚ディスプレイ用ソフトウェアの構成  
Fig.1 software configuration for haptic device

項目が必要となる。

1. 仮想物体の力覚特性を定めた力覚モデルとそれらをもとにフォースディスプレイに呈示させる反力を計算する力覚レンダリング
2. 物理法則などを司る法則エンジン
3. ユーザ同士やユーザと仮想物体などのユーザインタラクション

である（図1）。

それぞれについて詳しく述べていく。

#### 3.1 力覚モデルとレンダリング

仮想物体を記述するためには、その3次元形状を与えるほかに、重さや硬さといった力覚特性を定義する必要がある。これを与えるのが力覚モデルである。この力覚モデルをもとに、人間が感覚器官を通じて仮想物体を知覚できるよう力覚情報をフォースディスプレイの動作指令に変換する必要がある。これが、力覚レンダリングである。

力覚レンダリングには大きく分けると、サーフェイスレンダリングとボリュームレンダリングがある。サーフェイスレンダリングは、仮想物体の表面の硬さ、表面粗さ等を表現するものである。例えば、仮想物体の硬さを表現するために、力覚モデルとして、バネモデルを与えるとする（図2）。パラメータとしては、バネ定数を与えておく。力覚レンダリング部では仮想物体とフォースディスプレイの効果器との接触判定をし、力覚モデルに基づきめり込み量に比例した反力を呈示する。このパラメータを調節することで、固い物体から柔らかい物体までを表現することができる。

一方のボリュームレンダリングは、仮想物体内部の密

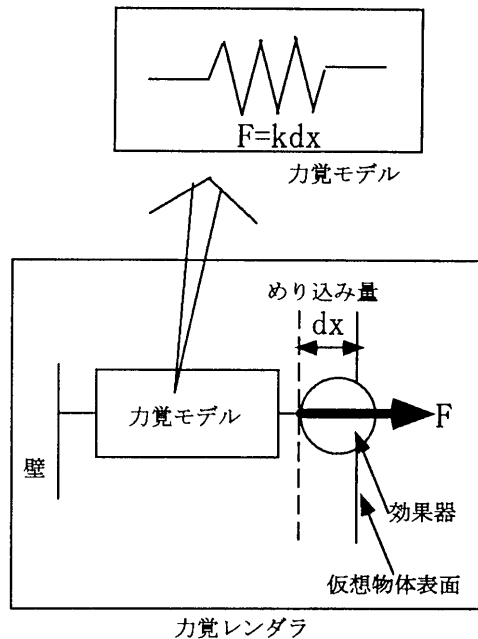


図2 力覚モデルと力覚レンダリングの関係  
Fig. 2 relationship between haptic model and haptic rendering

度や速度、応力分布等の属性を表現するものである。この場合、力とトルクをそれぞれの属性にマッピングする[15]。

力覚レンダリングとは仮想物体の状態をもとにフォースディスプレイにどのような反力を呈示すればよいのかを計算するもので、仮想物体の挙動に直接影響を与えるものではない。仮想物体の挙動に直接関わるのが、次節の法則エンジンである。

### 3.2 法則エンジン

我々の住む実世界は、様々な法則に支配されている。たとえば、物体が移動するときなどその挙動は物理法則によって予測することが出来る。ゴム膜や板などの変形の様子も運動方程式などを用いてある程度予測することが出来る。仮想世界でこれを実現するためには、これらの法則をプログラムとして与えなくてはならない。ところが、有限要素法などの数値シミュレーションは、計算に膨大な時間がかかる。人工現実感では、それらの計算を実時間、且つインタラクティブに行わなければならない。しかし、実時間を保証した数値シミュレーションの研究例は極めて少ない。したがってこれらの数値シミュレーションのプログラムを簡単に接続、あるいは交換できるような土台を作っておく必要がある。また、生物の発生や行動原理など、物理法則以外にも法則、原理が存在し、これらを実装することの出来るソフトウェア構成が求められる。

力覚レンダリングとの違いは、法則エンジンは仮想物体の挙動をシミュレートし、直接仮想物体のデータの更新をすることである。挙動計算をするときに外力など

を必要とする場合は、力覚レンダリングによって得られた力覚情報を元に、法則エンジンが仮想物体の情報を更新する形態をとる。力覚レンダリングと挙動シミュレーションを分けて考えることで、プログラムの理解が容易になるだけでなく、プログラムの汎用性を高めることができる。

### 3.3 ユーザ・インタラクション

仮想世界は、人間との対話があってはじめて有益なものになる。従来のグラフィックインタフェースは、マウスやキーボード、ウインドウ、アイコンなど2次元のユーザインタフェースであった。3次元仮想空間の空間性を生かしたユーザインタフェースは新たな可能性を有している。このような空間的なアイコンの操作を行う場合には、吸引力や拘束力等の呈示が操作性に大きく関わってくる。さらに人間と対話する機能を持った仮想物体の実装にも対応でき、仮想物体の創造や消去といった仮想空間の情報を管理する機構が必要となる。

また、仮想空間で作業をする人は一人とは限らない。多数数で、作業をすることで得られるメリットは大きい。この場合、仮想物体のデータを各ユーザ間で矛盾無く共有することの出来る枠組みが必要となる。

## 4. 力覚ディスプレイ用ソフトウェアの実装

### 4.1 VECSの基本構造

著者らは、仮想環境を構築するためのソフトウェアVECS( Virtual Environment Construction System )を開発している。このシステムは、以下の4つのモジュールからなる(図3)。

1. 法則エンジンを内蔵し、仮想物体の挙動を管理したりデータベースを管理するための本体部分
2. 各種フォースディスプレイを制御するためのデバイスドライバ
3. 力覚モデルや力覚レンダリング定義部
4. 様々なアプリケーションが格納されるアプリケーションプログラム

アプリケーションプログラムを取り替えることで容易に仮想環境を変更することができる。また、デバイスドライバを変更することで、異なる種類のフォースディスプレイと交換することもできる。

このシステムは、マルチプロセスのUNIXシステム上に実装されている。具体的にはSilicon Graphicsのグラフィックスワークステーション上で稼働しており、C++を用いて記述されている。視覚系のレンダラにはOpenGLを用いている。プログラムは、視覚系のためのVECS本体と、力覚系のためのアプリケーションプログラムの2つのプロセスに分かれている。アプリケーションプログラムは主に力覚情報の更新と仮想物体の特殊な挙動計算をおこない、アプリケーションプログラム

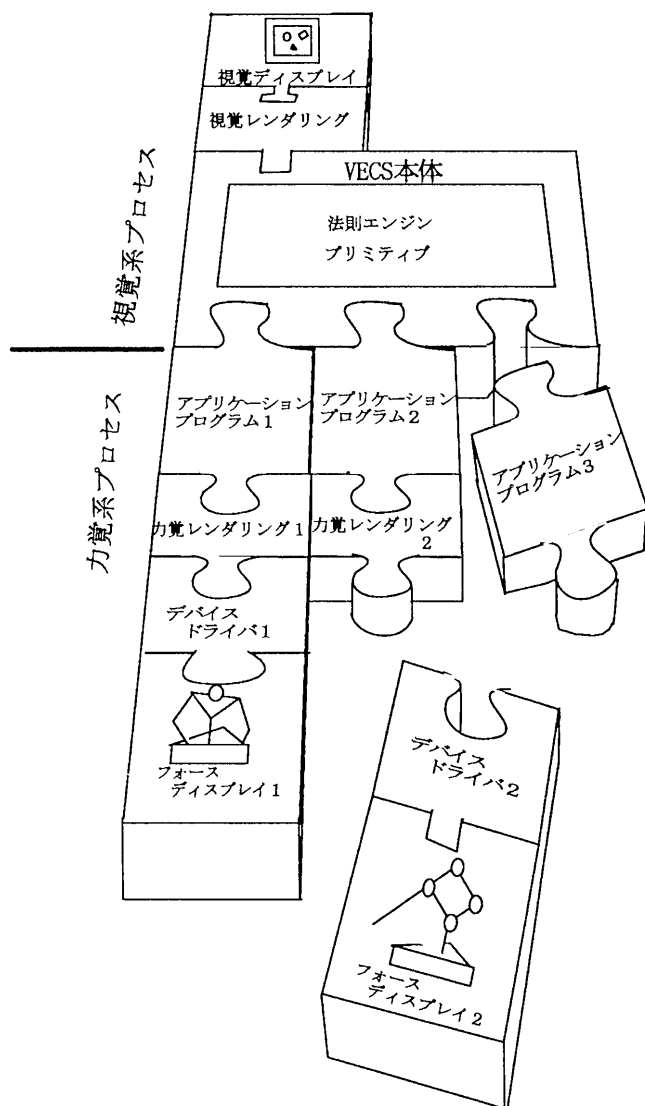


図3 システムの構成  
Fig. 3 System configuration

モジュール、力覚レンダリングモジュール、デバイスドライバモジュールの3つのモジュールとしてそれぞれ関数ライブラリの形で組み込まれている。VECS本体は、仮想空間のデータベースの管理、および映像の生成を行い、簡単な法則エンジン、プリミティブ視覚レンダリングモジュールからなる。2つのプロセスに分割することで、処理に時間のかかる視覚情報呈示処理が終了するのを待たずに、力覚情報の更新をおこなうことが出来る。

このようにするのは、人間にとって自然に知覚できる感覚情報の更新速度が、視覚に比べて力覚のほうがはるかに大きいからである。また、ソフトウェアとしてライブラリの形で機能を提供することも可能であるが、VECSはVECS本体単体でも動作するように設計されている。VECS本体が起動されていれば、ユーザが仮想空間に存在していなくても内蔵の法則エンジンにより仮想空間の情報が更新される。例えばある成長法則に従って

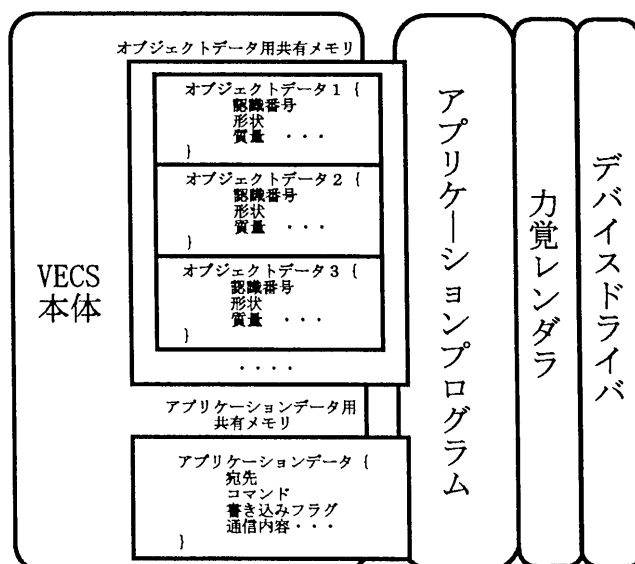


図4 データの構成  
Fig. 4 data configuration

成長する仮想の木を定義しておけばユーザがいなくても成長させることが可能である。

なおこれらのプロセスは、共有メモリを通してデータの通信を行っている。

VECS本体には、仮想物体のプリミティブが用意されており、球、直方体、円柱、円錐、ポリゴンメッシュ、手、ピンセット、ペン等をサポートしている。これらのデータはオブジェクトデータという構造体の形で全て共有メモリ上に実装される(図4)。オブジェクトデータには、仮想物体の認識番号、形状、質量、弾性係数、色、位置、速度、加速度、その他の情報を書き込む場所が確保されており、干渉チェックの結果や、仮想物体相互の連結情報も書き込まれる。一つの仮想物体に一つの共有メモリが割り当てられているので、仮想物体の創造と消去が容易に行える。

アプリケーションプログラムは、アプリケーションデータ構造体を用いてそのイベント情報を受け取る。これは、仮想空間にアプリケーションプログラムを接続した後の仮想物体の創造や消去の要求やその応答などをやりとりするための通信専用のデータ構造である。この中には、アプリケーションの認識番号、コマンド、宛先、読み込み書き込みフラグ、通信データ書き込み領域が確保されている。たとえば、アプリケーションプログラム側で、仮想物体創造の要求をコマンドに代入し、宛先をVECS本体にしてから書き込みフラグを出すとデータが送られる。すると折り返しVECS本体からアプリケーションに新たに創造された共有メモリの認識番号が送られてくる。それをもとにアプリケーションが共有メモリにアクセスすることで、細かい仮想物体の属性を読み書きできるようになる。

また、複数のアプリケーションを VECS に接続することで、複数のフォースディスプレイを接続することが出来、協調作業のプログラムを組むことも容易に出来る。これは、VECS 本体のデータベースを常に参照でき、仮想空間でのイベント（ユーザの増減や仮想物体の増減等）をアプリケーションデータによって通知する仕組みにより、協調作業環境を構築するにあたっての情報の整合性を容易にとることができるからである。

VECS のユーザインタフェースには、仮想世界に命令を与えるためのボタンとスライドバーが用意されている。フォースディスプレイの動きと連動した仮想物体が、ボタン等に触れるとその中心に引き込まれるような反力がユーザに与えられる（図 7）。この機能によって、空中に浮かぶボタン等の操作がしやすくなっている。

#### 4.2 各モジュールの機能

各部の論理的機能のについて述べる。

力覚レンダリング部は、デバイスドライバからデバイスの位置姿勢等をデバイス固有の座標系データとして受け取る。その後 VECS 本体内に保管された仮想物体のデータに直接アクセスし、力覚モデルから力や速度、位置等の情報を得、デバイスに依存しない仮想空間の座標系表現の力ベクトルを計算する。その結果をデバイスドライバに通知する。

デバイスドライバは、デバイスのセンサーデータを取り込み、デバイス固有の座標系の位置姿勢等のデータに変換する。そのデータを力覚レンダラに出力する。また、力覚レンダラからの入力として、仮想空間での力ベクトル等を受け取り、デバイス固有の力ベクトル等に変換する。そのデータを基にデバイスのアクチュエータを制御する。

アプリケーションプログラムモジュールは、仮想空間のレイアウトを決めるために VECS に仮想物体の創造や消去の要求を出す。また、仮想空間のイベント情報を VECS から受け取り、共有メモリの接続、切り放し等を行う。ユーザインタラクションに関しては、例えば、データを入力するためのスライドバーやボタン等の仮想物体を触ったときにどのような反力を呈示するか、どのようなデータを変化させるのかといった事についての定義をおこなっている。

VECS 本体では、物理法則による仮想物体の挙動の計算をおこなっている。現在は重力、衝突、粘性をサポートしている。VECS 本体に内蔵された法則エンジンを使うことで、アプリケーションプログラム側は新たな物理法則や特殊な法則だけを実装してやればよいことになる。また、仮想空間のデータを基に仮想空間の映像生成をおこなっている。このほかに、アプリケーションプログラムとの通信をおこなう。

プロセス同士の通信路の確立は、まず一般に公開され

ている初期通信共有メモリを通してアプリケーションプログラムが、VECS へ仮想空間への参加を通知する。折り返し VECS からは、アプリケーションデータ用共有メモリの認識番号（各プロセスに一つ）が返されてくる。以後はこのデータを使って、仮想物体の創造や消去の要求やその応答などをやりとりする。このようにすることで、複数のアプリケーションを接続することもできる。

#### 4.3 仮想物体のデータフォーマット

また、VECS はデータフォーマットとして VRML ver1.0 をサポートしている。VRML では、球や、円柱、直方体といった形状プリミティブの他に、平行移動や、回転移動、色属性、他の WWW サーバへのポインタなど様々なプリミティブがあり、それらが階層構造を持ったノードとして定義されている。VECS のオブジェクトデータに翻訳するときには、これらをそのまま扱えるよう、VRML オブジェクトというものを定義している。VRML の基本構造として、認識番号、VRML オブジェクトであることを示すフラグ、どの種類のプリミティブなのかを示すフラグ、親オブジェクト番号、子オブジェクト番号をもたせている。これにそれぞれのプリミティブ特有のデータ構造を付加する。干渉チェックなどの処理をおこなう場合は、親オブジェクトから順に子オブジェクトへとスタックを積みながら計算をする。干渉チェックは、デバイスとの干渉球の干渉でおこなう方法や、仮想物体表面を平面で近似しその平面とのめりこみ量で判断する方法など、プリミティブによって使い分ける。干渉チェックは早い情報更新が必要なので、アプリケーションプログラムで計算を行っている。http サーバとの通信には、telnet コマンドを使用して行っており、世界中の VRML ファイルをネットワークを介して取り寄せ仮想空間に展開することが出来る。

#### 4.4 実行速度

現在、VECS は SGI の Indigo2 MAX IMPACT , Indy R5000, Onyx i-station 上で稼働している。フォースディスプレイとは RS232c を用いて各関節の角度と各モータの目標トルクをやり取りしている。次節の実験と同じ環境での実行速度は、IMPACT, Onyx 上で視覚系 60Hz, 力覚系 50Hz, Indy 上では、視覚 5Hz, 力覚 25Hz となっている。力覚と視覚の更新速度がほぼ同じになっているのは、フォースディスプレイとの通信速度がワークステーションの初期設定により、固定されていたことが原因である。力覚は数十から数 KHz の更新速度が必要といわれている。本システムでは非常に固い壁や非常に細かい凹凸などを除いてほとんどの仮想物体を表現することが出来る。また、視覚系の速度は仮想空間の映像が複雑になるにつれて急激に遅くなるので、IMPACT や Onyx で 2 つのプロセスに分けることは無意味ではない。

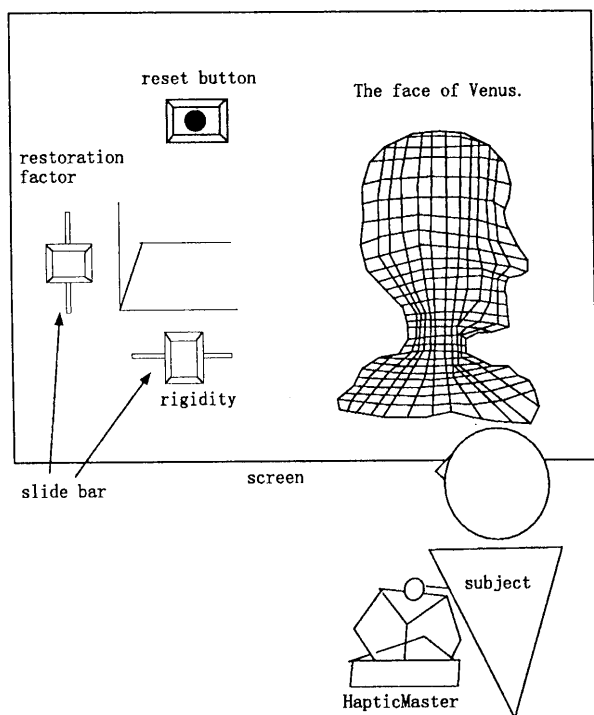


図5 実験環境  
Fig. 5 environment of experiment

## 5. 力覚帰還環境の操作性評価

### 5.1 実験環境

本研究では、VECSを用いて構築した力覚帰還環境の操作性の評価をおこなった。これまで、著者らは実験室内で、以下のような実験をおこない、力覚フィードバックの有効性を示してきた。仮想物体をある形に変形する際の作業時間を力覚の有無で比較したり [16]、遠隔地同士をインターネットを介して接続し、時間遅れのある環境においても力覚呈示をすることで、教示作業を正確に行えることを示してきた [17]。これらは全て実験室内のことで、実用化を考えると多人数の被験者を用いた調査をおこなう必要がある。このため次のような実験をおこなった。

多数の多様な被験者に力覚帰還型の仮想空間を操作してもらった。そのときの様子をいくつかの項目に基づいて観察し、それらを統計的に処理することで評価をおこなう。

計算機はSGI Indy R5000、フォースディスプレイとして日商エレクトロニクスのHapticMasterを使用し、60インチ相当のスクリーンに仮想空間の映像を呈示した。この仮想空間には、ミロのビーナスの形をした自由曲面が存在する。これはVRMLで記述された800ポリゴンのモデルであり、フラットシェーディングで描画されている(図5)。この自由曲面はケルビンモデル[18]によるバネとダンパーによって各格子点をつないだ曲面で表現している(図6)。フォースディスプレイで曲

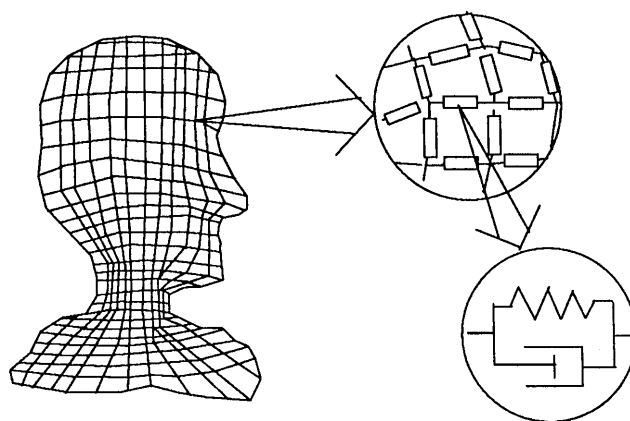


図6 バネ・ダンパーモデル  
Fig. 6 connection model of each lattice of free-form surface

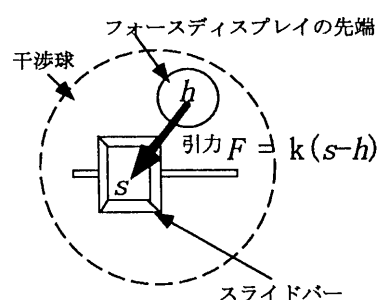


図7 ボタンやスライドバーからの反力  
Fig. 7 applied force from slide bar or button

面の表面を押し込むと、押し込み量に比例して押し返す反力が呈示され、表面をなぞっているような感覚を与える。この反力に応じてバネ・ダンパーモデルに基づき曲面の変形が起こる。大変形が起こった場合、降伏点を超えた格子点については塑性変形が起こる。左側にある2つのスライドバーで、自由曲面の力覚的特性(バネ定数と降伏応力)を変化させる。バネ定数(rigidity)を変化させると自由曲面の硬さが変わる。降伏応力(restoration factor)を大きくすると、もとの形に戻りやすくなる。また、スライドバーやボタンに触れたときは以下のように反力を計算する(図7)。スライドバーやボタンの干渉球の内側に手先が入ると触っているとみなし、バネの力覚モデルに基づいたボタン中心に引き込む反力を呈示する。さらに、フォースディスプレイに付けられたタクトスイッチを押すと仮想物体を移動することができる。自由曲面をつかむと重力による重さが呈示される。

### 5.2 観察項目

観察した項目は以下の4つである。(1) 力覚帰還を知覚できたか。(2) 3次元アイコンの使い方が理解できたか。(3) バネ定数を変化させたとき、自由曲面に触ったときの硬さの変化を理解しているか。(4) 降伏応力を変化させたとき、自由曲面の復元量に変化することを理解できたか。

## 矢野・岩田：力覚帰還型仮想環境構築ソフトウェア

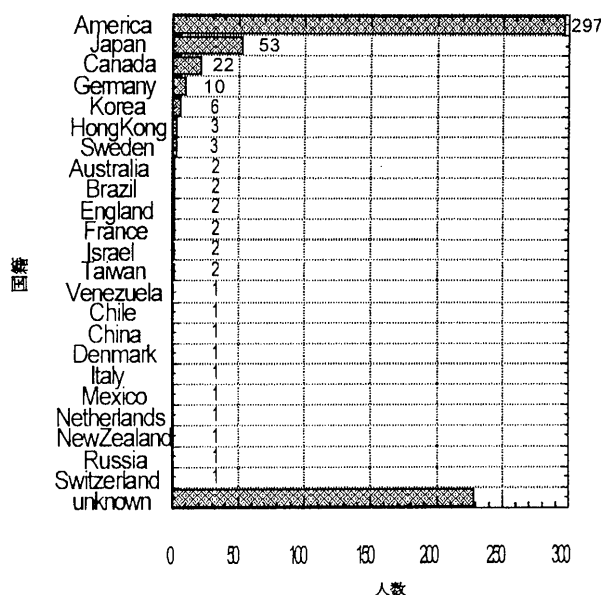


図 8 被験者の国別人数

Fig. 8 various nationality of subjects

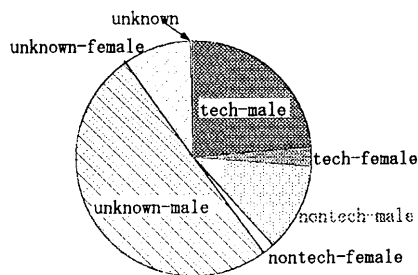


図 9 被験者の構成

Fig. 9 percentage of subjects by occupation

それぞれ 4 項目について、知覚できた (yes)、出来ない (no)、半分くらいわかっている (maybe)。触らなかった (untouch)。のどれにあたるかを著者らが観察し、記録を残した。各人がつけているネームプレートから、わかる範囲で職業も記録した。

実験は 1996 年 8 月にニューオリンズで 5 日間にわたって行われた SIGGRAPH96 でのデモ展示と同時にに行った。操作方法の説明を行いながら実際に触ってもらい、647 人 (うち女性 88 人) のデータを採取した。参考までに被験者の国別人数を図 8 に、職業別の構成を図 9 にのせる。実験室での実験とは違い、多人数多様な被験者層であることがわかる。実験の結果を円グラフにして図 10 にのせる。ここでの unknown は、記録漏れなど何らかの理由で記録が残っていないことを示す。

前述した項目 (1) に関しては 637 名 (98%) が知覚できた。知覚できない人が 3 名いた。(2) については、操作した人が全体の 69% で、その内の 99% の人が操作法を理解していた。(3) については、操作した人が全体の 61% で、その内の 74% の人が硬さが変化していることを理解していた。(4) については、操作した人が全体の

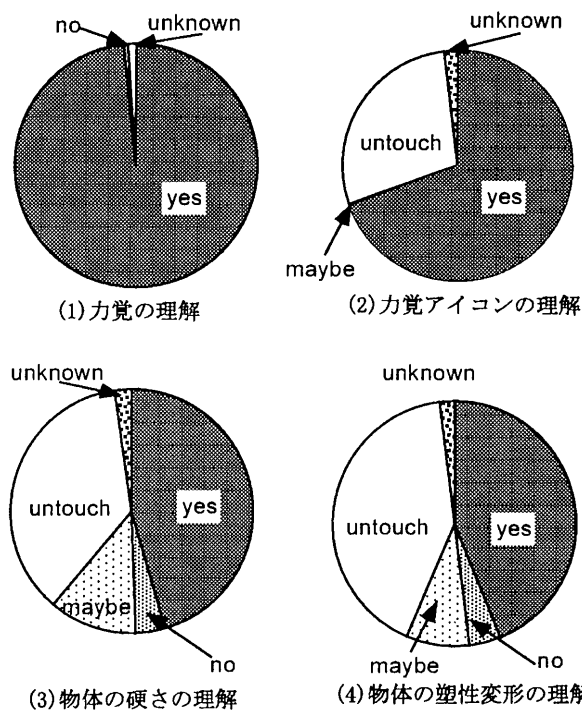


図 10 全体の実験結果

Fig. 10 Result of experiment by all subject

56% で、その内の 77% の人が復元量に変化していることを理解していた。

## 6. 考 察

力覚フィードバックを知覚できなかった人は 3 名いた。このうち 2 名については十分にフォースディスプレイに触る時間がなかったようで、すぐ帰ってしまった人である。残りの一名は、呈示していた CG の奥行き情報が余りよく出ていなかった (単眼視、影なし、間接指示) ため、どの辺りを触っているのか理解することが出来なかったようである。3 次元のアイコンについては、スライドバーやボタンなど引力を呈示してやることで、位置などの識別が容易になることが、被験者の感想からもわかった。力覚アイコンに触らない人が 31% いたことについては、アイコンがただのボタン型で、それを触ると何が起こるのかを予感させるものがなかったため、あまり関心を持たせることが出来なかったことと、フォースディスプレイに初めて触れる人が多く自由曲面を持って移動するとき呈示される重さに関心が集まり、他の機能を試すまでに至らなかったことが原因と考える。

4 章の実験で、更に技術系 (180 人)、非技術系 (89 人) に分けてデータを整理した (図 11) (図 12)。この結果、(1) に関しては特に差はなかった。(2),(3),(4) については、理解度が明らかに技術系の方が高い。細かい力覚特性については、それを理解するためのバックグラウンドとなる知識がある程度は必要であるためと考える。また、CAD 等のソフトウェアエンジニアがかなりの人数いた

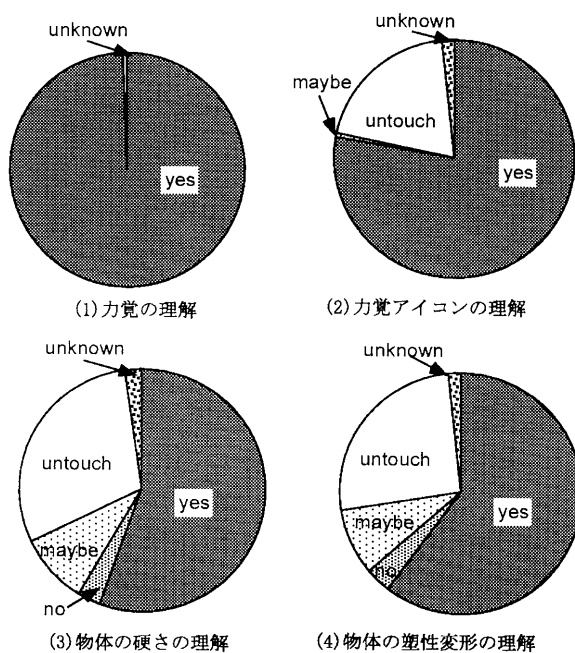


図 11 実験結果 技術系  
Fig. 11 Result of experiment by technologist

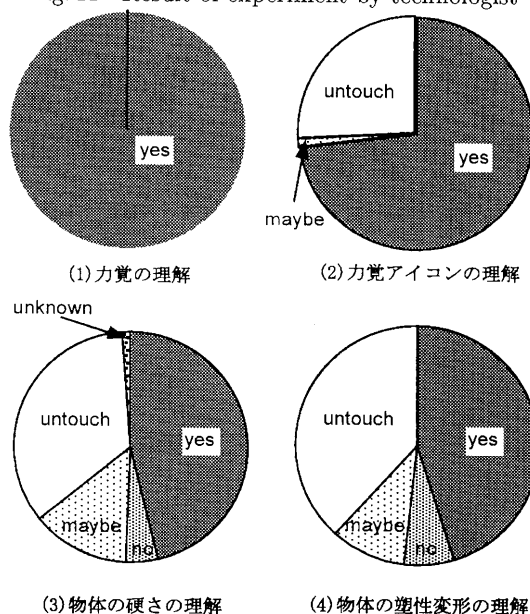


図 12 実験結果 非技術系  
Fig. 12 Result of experiment by nontechnologist

こともアイコンを操作する人数の差に影響している。

また、性別ごとに整理すると(図 13)(図 14)、(1)に関しては特に差はなかった。(2),(3),(4)については、理解度が明らかに男性の方が高い。これについては、実験をしている最中の被験者の意見から、フォースディスプレイの外観に問題があることがわかった。ギヤが見えていたり、まれに発振を起こすなど、いじっていて壊してしまうのではないかとこの恐れを抱かせてしまう。これがもともと、力覚アイコンなどを触る気力が萎えてしまう事がわかった。

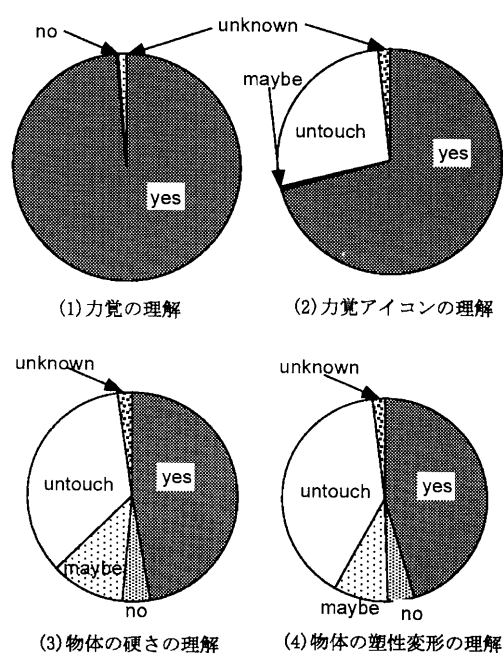


図 13 実験結果 男性  
Fig. 13 Result of experiment by male

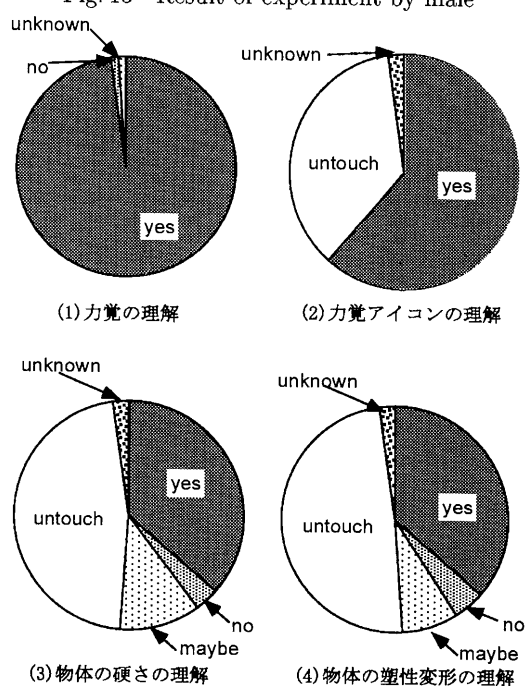


図 14 実験結果 女性  
Fig. 14 Result of experiment by female

これらの結果から、力覚は、形状の理解、操作に役立っていることがわかる。また、フォースディスプレイ等は、外観が壊れにくいイメージを与える必要がある。たとえば、ギヤやモータ、配線等は極力見えないようにし、発振など極力起こらないようにすることで操作者の予測とは違う動きをしないようプログラムを組む必要があることがわかった。



## 7. ま と め

力覚帰還環境を構築するための基本構成について述べた。このソフトウェアに必要な項目は、(1) 仮想物体の形状の他に力覚特性を定めた力覚モデルとそれらをもとにフォースディスプレイに呈示させる反力を計算する力覚レンダリング、(2) 物理法則などを司る法則エンジン、(3) ユーザー同士やユーザと仮想物体などのユーザインタラクション、の3つである。全てを満たす実装形式の一つとして、VECSの実装例を示し、多人数の被験者を観察するという方法で、VECSによって構築した力覚帰還環境の操作性についての評価を行った。その結果、力覚はほぼ全員が知覚できること、力覚アイコンについてもほぼ全員が使いこなすことが出来ること、力覚アイコンに付加された機能も74%以上の人実際に触ってみることによって理解できることがわかり、VECSが有効に動作していることが分かった。

今後の課題としては、VRMLのプリミティブの充実、仮想空間をいかにわかりやすく提示すればよいのかの演出手法の開発、これらのソフトウェアを使ったコンテンツづくりが挙げられる。

## 8. 謝 辞

実験を行うに際して筑波大学大学院理工学研究科の市ヶ谷敦郎氏に協力をしていただいた。ここに感謝の意を表したいと思う。

## 参 考 文 献

- [1] 広田光一, 木島竜吾: パーチャルリアリティ・ソフトウェア, 仮想現実学への序曲, 共立出版, 132/142 (1994)
- [2] 柳田, 館: クラスを用いた仮想環境構成法, 電気学会論文集C分冊, 平成7年2月号, 236/244 (1995)
- [3] WorldToolKit Windows USER'S GUIDE, SENSE8
- [4] ViScape: <http://www.superscape.com>
- [5] dVS: <http://www.division.com/3.pro/software/index.html>
- [6] Mark, Randolph et al: Adding Force feedback to Graphics Systems: Issues and Solutions, COMPUTER GRAPHICS PROCEEDINGS, 447/452 (1996)
- [7] GHOST General Haptics Open Software Toolkit, Brochure of SensAble Technologies Inc. (1996)
- [8] 矢野 博明, 中川 隆志, 岩田 洋夫: フォースディスプレイを用いた仮想環境における協調作業, SIG Note, Vol.93.No34(1993)
- [9] Hiroo IWATA, Hiroaki YANO: Artificial Life in Haptic Virtual Environment, Proceedings of ICAT'93, 91/96 (1993)
- [10] 矢野 博明, 岩田 洋夫: 人工生命に対応した仮想環境構築システム, 第9回ヒューマンインタフェースシンポジウム論文集, 129/138 (1993)
- [11] 矢野 博明, 岩田 洋夫: 力覚帰還型仮想環境における共同作業, IPSJ SIG Notes Vol.94, No.59, 31/34 (1994)
- [12] Hiroo IWATA, Hiroaki YANO: Interaction with Autonomous Free-form Surface, Proceedings of ICAT'94, 27/32 (1994)

- [13] 矢野 博明, 岩田 洋夫: 自律的自由曲面を用いた仮想環境における協調作業, 電気学会論文集C編 115 巻 2 号, 245/252 (1995)
- [14] Hiroaki YANO, Hiroo IWATA: Cooperative Work in Virtual Environment with Force Feedback, Proceedings of ICAT/VRST '95, 203/210 (1995)
- [15] Iwata and Noma: VOLUME HAPTIZATION, PROCEEDINGS OF THE IEEE 1993 SYMPOSIUM ON RESEARCH FRONTIERS IN VIRTUAL REALITY, 16/23 (1993)
- [16] 矢野 博明, 岩田 洋夫: 力帰還型仮想環境における協調作業, 計測自動制御学会論文集 Vol.31, No.9, 1495/1501 (1995)
- [17] 矢野 博明, 岩田 洋夫: フォースディスプレイを用いた分散型仮想環境における協調作業, 第11回ヒューマン・インタフェース論文集, pp389-394 (1995)
- [18] 西岡 隆: 構造振動解析, pp47-48, 培風館

(1996年11月8日受付)