力覚ﾌｨｰﾄﾞﾊﾞｯｸ機能を有する VRML2.0ﾌﾞﾗｳｻﾞの開発

呉寧[1)], 安田孝美[2)], 横井茂樹[3)]
1)名古屋大学　人間情報学研究科
(〒464-01 名古屋市千種区不老町, wuning@yokoi.gs.human.nagoya-u.ac.jp)
2)名古屋大学　情報文化学部
(〒464-01 名古屋市千種区不老町, yasuda@info.human.nagoya-u.ac.jp)
3)名古屋大学　情報文化学部
(〒464-01 名古屋市千種区不老町, yokoi@toriwaki.nuie.nagoya-u.ac.jp)

概要: VRMLﾌﾞﾗｳｻﾞに力学ﾌｨｰﾄﾞﾊﾞｯｸを付加した試作ｼｽﾃﾑを紹介する。力学ﾌｨｰﾄﾞﾊﾞｯｸつきﾌﾞﾗｳｻﾞと伝統的なﾌﾞﾗｳｻﾞとは、実現上においてかなり違う点が存在する。力覚装置を使用して、安定なﾌｨｰﾄﾞﾊﾞｯｸ力を得るために、非常に高い更新速度で反力ﾊﾟﾗﾒｰﾀの計算を行う必要がある。本文は、これについての一つの対処策として、筆者らが開発した **Force Field Rendering** 技術を紹介する。

ｷｰﾜｰﾄﾞ: *Force Field Rendering , VRML2.0, PHANToM*

VRML 2.0 browser with A Force Feedback Device

Ning WU, Takami YASUDA and Shigeki YOKOI

Abstract: This paper describes a system which uses a force feedback device -- PHANToM to manipulate the VRML 2.0 virtual environment. The implementation of VRML browser with force feedback is quite different from that of the ordinary VRML browser. In order to obtain steady force feedback, haptic interface requires very high update rate. Our system uses a method which we call it **Force Field Rendering** to enable the PHANToM haptic device to manipulate the VRML 2.0 virtual world—one get real time force feed back as if there were real objects in place! At the same time, the manipulated world will be displayed on Windows 95 using Direct3D or on SGI workstation using Open Inventor.

Key Words: *Force Field Rendering , VRML2.0, PHANToM*

1. はじめに

VRML は 3 次元仮想空間をｲﾝﾀｰﾈｯﾄ上に実現する技術として注目されている。すでにVRML1.0 の仕様に基づいてﾊﾟｿｺﾝにおいてもﾌﾞﾗｳｻﾞが開発され、我々にとって身近な技術になっている。VRML は仕様の拡張が検討され、96 年 8 月 4 日に VRML2.0ﾌｧｲﾅﾙ仕様[1]が確定された。これは動きや形状変形の表現も含み、より高度な仮想空間記述を可能にしてい

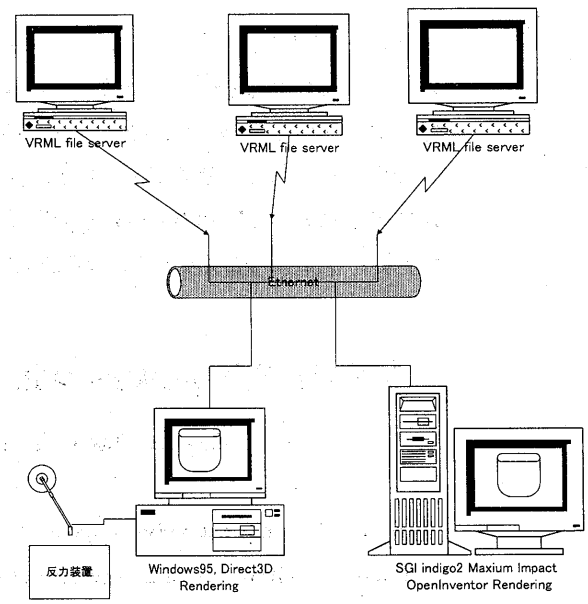
る。ところで、VRMLﾌﾞﾗｳｻﾞは映像(音声も含む)を利用したﾌﾞﾗｳｻﾞであり 3 次元仮想空間を視覚(および聴覚)により体験可能にするものであるが、仮想空間との体験には力覚も重要なはたらきをしている。本研究では高精度の商用力覚ﾌｨｰﾄﾞﾊﾞｯｸ装置 PHANToM を利用して、VRMLﾃﾞｰﾀを視覚および力覚で体験できるｼｽﾃﾑの開発を試みた。本研究は、世界中でこれから構築が進むと思われる VRML2.0 仕様のﾃﾞｰﾀﾌｧｲﾙを解析して、3 次元表示するとともに力覚ﾌｨｰﾄﾞﾊﾞｯｸ装置により力覚で体験するためのｼｽﾃﾑである。

力学ﾌｨｰﾄﾞﾊﾞｯｸつきﾌﾞﾗｳｻﾞと伝統的なﾌﾞﾗｳｻﾞとは、実現上にかなり違う点がある。特に力覚装置--Phantom を使用して、安定なﾌｨｰﾄﾞﾊﾞｯｸ力を得るために、1 秒間に 800 回以上、装置に反力ﾊﾟﾗﾒｰﾀを渡さなければならない。つまり、反力の計算は 800Hz のｽﾋﾟｰﾄﾞで実行すべき、普通のﾌﾙｽﾋﾟｰﾄﾞｱﾆﾒｰｼｮﾝの 30Hz のｱﾌﾟﾃﾞｰﾄﾚｰﾄよりはるかに厳しい条件である。本文は、おもにこれについての一つの対処策として筆者らが開発した新しい **Force Field Rendering** を紹介する。

2. ｼｽﾃﾑ構成

ｼｽﾃﾑを構成するﾏｼﾝは力覚ﾌｨｰﾄﾞﾊﾞｯｸ装置を制御する Windows95ﾏｼﾝと仮想空間を表示するためのｸﾞﾗﾌｨｯｸﾜｰｸｽﾃｰｼｮﾝ SGI Indigo2 により構成されている。

Windowsﾏｼﾝのﾌﾞﾗｳｻﾞから VRML2.0 のﾌｧｲﾙをﾀﾞｳﾝﾛｰﾄﾞする。これを構文解析してｵﾌﾞｼﾞｪｸﾄの直接的記述を得る。この記述を利用して、記述内容を 3 次元表示する。高速なｲﾝﾀﾗｸｼｮﾝのために CGﾜｰｸｽﾃｰｼｮﾝにﾃﾞｰﾀを転送して表示する。CGﾜｰｸｽﾃｰｼｮﾝで表示したり、ﾅﾋﾞｹﾞｰﾄするとともに、Windows95ﾏｼﾝで **Force Field Rendering** する。**Force Field Rendering** したｼｰﾝは PHANToM により力覚で体感することができる。



3. ｿﾌﾄｳｴｱ開発の方針、特徴

ｼｽﾃﾑに必要な諸機能のｿﾌﾄｳｴｱを単に実現するということではなく、ｼｽﾃﾑの拡張性、汎用性、他のﾊｰﾄﾞｳｴｱへの移植性、最新技術の導入などを考慮して開発を行った。また、ｲﾝﾀｰﾈｯﾄ上のﾊﾟﾌﾞﾘｯｸﾄﾞﾒｲﾝのｿﾌﾄｳｴｱﾗｲﾌﾞﾗﾘ(例えば、GNU Flex/Bison、SGI VRML2.0 Final Draftｻﾝﾌﾟﾙﾊﾟｰｻｰ[2][3]、C++Standard Template Library[4])を有効に活用して効率的な開発を行った。

ｿﾌﾄｳｴｱはいくつかの独自のﾌｧﾝｸｼｮﾝを持つ C++ｸﾗｽﾗｲﾌﾞﾗﾘとして構成している。ｸﾗｽﾗｲﾌﾞﾗﾘ間は、わずかの標準化しているｲﾝﾀｰﾌｪｰｽにより接続しているため、将来特定のｸﾗｽﾗｲﾌﾞﾗﾘを拡張、変更しても他の部分はほとんど影響なく利用可能になっている。

また、PHANToM 装置の反力ﾊﾟﾗﾒｰﾀを 800Hz 以上更新すべき問題についても、Force Field Rendering という独自の方策で対処した。

4. 開発ソフトウエアの内容

4.1 VRML2.0ファイルのパーサー

VRML2.0ファイルを構文解析して、VRMLオブジェクトの直接的な表現を求める。SGI の VRML2.0 Final Draftサンプルパーサーに基づいて作成した。このサンプルパーサーは VRML2.0 の文法のチェックを行う機能のみであるが、これを拡張して、シーンの直接的記述を得るソフトウエアを開発した。パーサーは UNIX のコンパイラ開発ツールの lex/yacc[6]の GNUバージョンである Flex と Bison により開発した。VRMLノード構造、シーングラフなどのデータ構造の処理は STL(Standard Template Library)により行った。

パーシングしたデータは標準ノード、汎用ノードとグループノードオブジェクトの集まりとして表現され、シーンを直接的に表現したものになっている。

4.2 3次元表示のためのクラスライブラリ

PHANToM装置は Windows95マシンに接続され、制御される。 Windowsマシンでは 3D 表示の能力が不足するため、Windowsマシンで反力装置の制御を行い、SGI の CGワークステーションにより 3D 表示を行う設計をとっている。しかし、Windowsマシン制御を適切に行うためには Windowsマシンでも 3D 表示することが必要であり、両方のマシンで 3D 表示機能を開発した。3D 表示機能は開発用ライブラリとして、windowsマシンでは Direct3D、Indigo2 では OpenInventor を使用して開発した。

4.3 通信クラスライブラリ

ネットワークによるダウンロードやマシン間の通信を行うため、 C++の入出力クラスライブラリ iostream を拡張し、ストリームソケット、およびデータグラムソケットをカプセル化することにより、ソケットの知識がなくても iostreamライブラリのインターフェースと同一形式でアクセスできるようにした。

4.4 反力計算関連クラスライブラリ

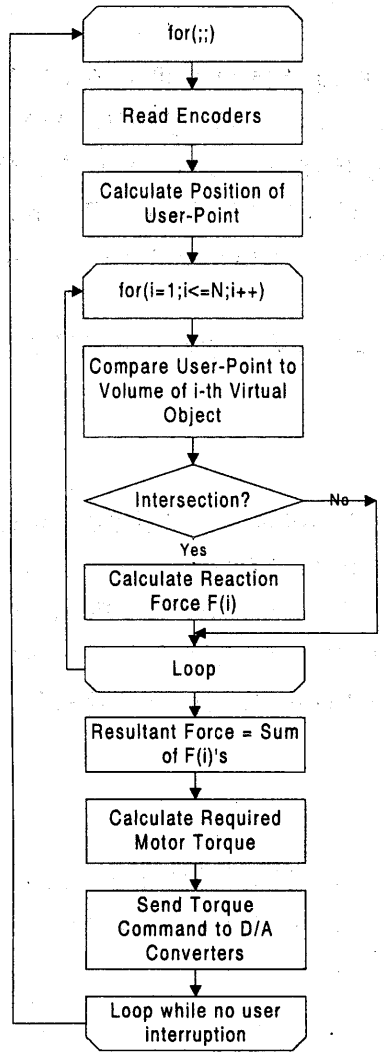
反力装置 PHANToM は、MIT の AI 研究所において開発されたものである。マニピュレータの先端部分に指入れが装着されてあり、ここに指を入れて移動すると、その XYZ 座標値と移動速度が計測できる。このデータはコンピュータに入力され、コンピュータにより指の運動が仮想環境において行われたものとして計算して、仮想物体からの反力などを計算してマニュピュレータにより指先に与える。

5. 反力計算関連クラスライブラリ

反力計算関連クラスライブラリの内容を紹介する。

5.1 反力計算の過程

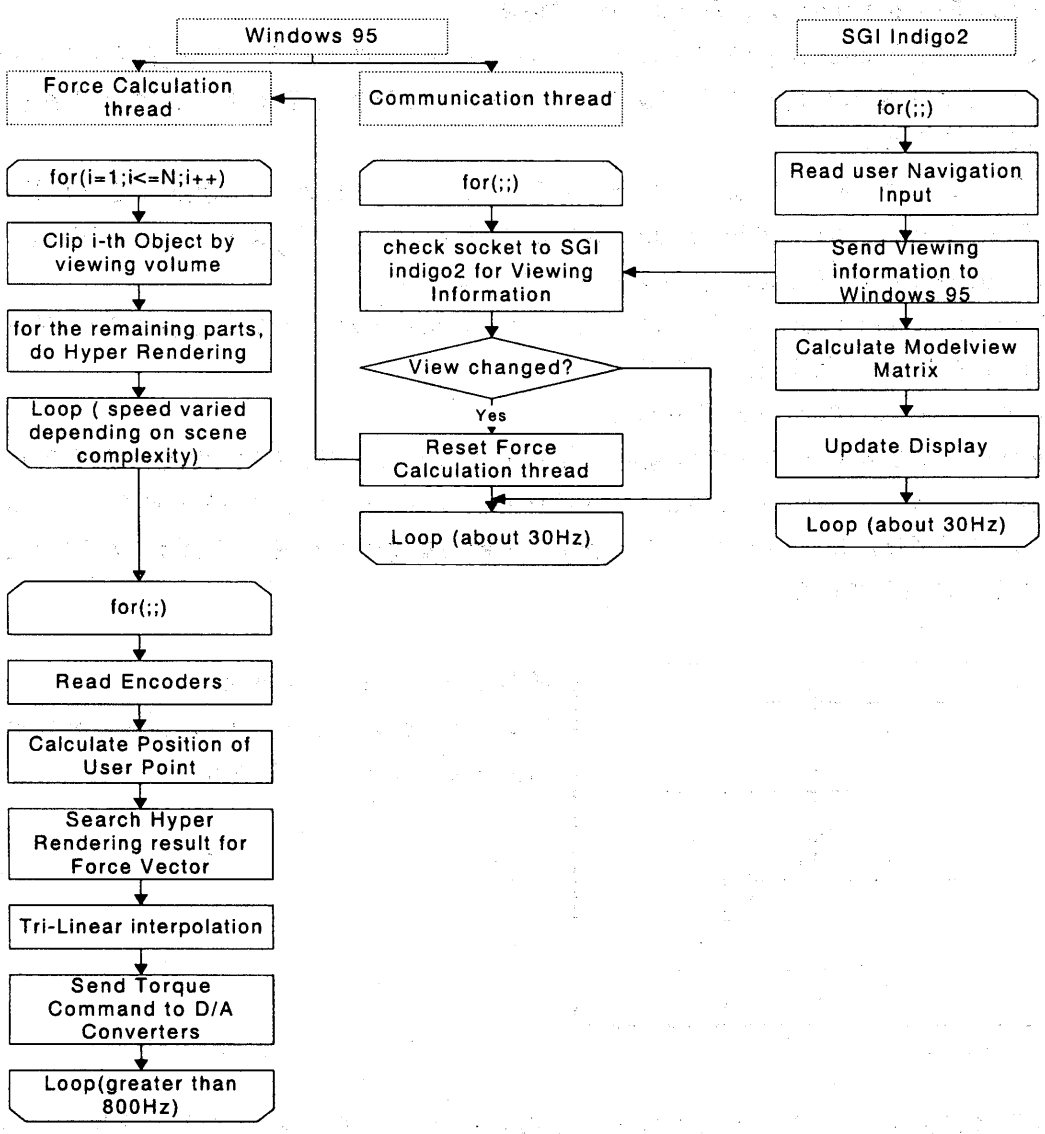
PHANToM 装置に対し、1 秒間 800 回以上、反力用の制御データを送らないと安定な反力が得られない。これは画面上のレンダリング (フルスピードアニメーションは 30Hz)より、はるかに厳しい。



反力計算の標準的な過程は、[5]によると、左図のとおりである。
外側のループを 800Hz で実行することは、多少でも複雑なワールド(千ポリゴン以上)になれば、現在のパーソナルコンピューターでは処理能力が不足してしまう。これに対して、BSPツリーや OBBツリーなどアクセラレーション技術の利用を検討したが、この問題に十分対処できるものではなかった。

この問題に対する適切な解決方法は難しいと考えていたが、ある日次のことを思い出した。レートレシングがおそいが、**プリ計算**すれば、フルスピードで photo-realistic アニメーションができる。プリ計算を導入すれば、複雑なワールドも手で体感できる。

まず、プリ計算を導入したプログラムの流れから紹介しよう。各クラスライブラリは Multithread およびネットワーク対応として設計されているので、ネットワークで通信により、Windowsマシンでの反力操作と Indigo2 での 3 次元表示を並列的に実行することが可能である。
SGIワークステーションがナビゲーション用、Windowsマシンは反力計算用に用いられる。



5.2 Force Field Rendering とは

基本的な発想はｼﾝﾌﾟﾙなものである。標準的な反力計算の過程において一番時間かかるのは、ﾕｰｻﾞｰﾎﾟｲﾝﾄを使って各ｵﾌﾞｼﾞｪｸﾄと交差判断して各ｵﾌﾞｼﾞｪｸﾄからの反力を計算するところである。しかし、この計算は 800Hz で行う必要はない。それを内部ﾙｰﾌﾟから外して、ﾌﾟﾘ計算して、PHANToM のﾃﾞﾊﾞｲｽ座標系の各点の反力ﾍﾞｸﾄﾙを生成してﾃｰﾌﾞﾙに記憶しておく。この結果、800Hz で更新計算するための処理は、主としてﾃｰﾌﾞﾙからの検索を行えばよいことになり、計算が少ないから、ﾜｰﾙﾄﾞの複雑さにほとんど依存せず実行できる。PHANToM のﾃﾞﾊﾞｲｽ座標系は三次元なので、普通の Display の二次元ﾃﾞﾊﾞｲｽ座標系へのﾚﾝﾀﾞﾘﾝｸﾞとの区別のために、Force Field Rendering という。

普通ﾚﾝﾀﾞﾘﾝｸﾞと Force Field Rendering の比較

	from	to	結果次元	生成内容
普通	ｵﾌﾞｼﾞｪｸﾄ座標系	ｽｸﾘｰﾝﾃﾞﾊﾞｲｽ座標系	2	点のｶﾗｰﾍﾞｸﾄﾙ
Force Field	ｵﾌﾞｼﾞｪｸﾄ座標系	PHANToMﾃﾞﾊﾞｲｽ座標系	3	点の反力ﾍﾞｸﾄﾙ

5.3 Force Field Rendering のためのﾃﾞｰﾀ構造

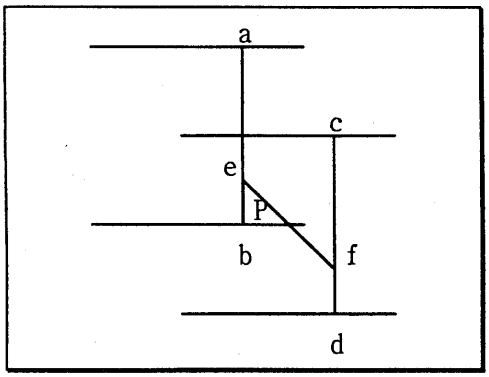
Force Field Rendering の結果の出力において、直接的な方法は二次元画像のﾌﾚｰﾑﾊﾞｯﾌｧのように三次元配列を使うことである。しかし、下の計算の示すようにこれはあまり現実なものとは言えない。

PHANToM のﾌﾟﾛｸﾞﾗﾑによく使う BIG BOXﾙｰﾁﾝ、つまり PHANToMﾃﾞﾊﾞｲｽ実測できる座標範囲は約 3000<x<6000 -4000<y<4000 -6000<z<-2000 である。反力ﾍﾞｸﾄﾙは x, y, z 各 16bit である。それで、Force Field Rendering 結果をそのまま記憶すると、

(16+16+16)*3000*8000*4000/8=576*10^9 Byte = 576GB が必要である。PC のﾒﾓﾘ量は 32MB-128MB が主流なので、何千倍も必要である。

解決策として二つの方法を使用している。

一番目は、3000*8000*4000 の解像度で保存ではなく、300*800*400 の低い解像度で保存して、保存していない点の反力ﾍﾞｸﾄﾙは Tri-Linear interpolation で補間する方法である。最適なｻﾝﾌﾟﾘﾝｸﾞ解像度については、個人の触覚の主観的な差違が存在し、実験ﾃﾞｰﾀが不十分なため、まだ求められていない。



補間 1:a, b, c, d を求める。
補間 2:a, b から e を c, d から f を求める
補間 3:e, f から P を求める。

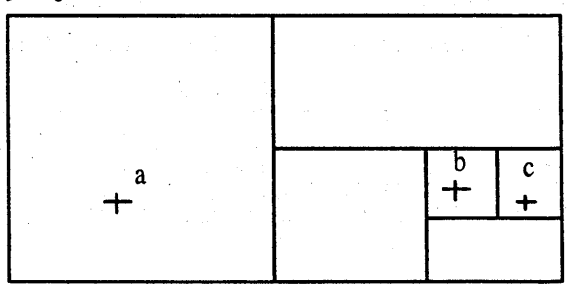
二番目は、PHANToMﾃﾞﾊﾞｲｽ空間のほとんどの点の反力が 0 であることに注目し、配列構造ではなく、ﾂﾘｰ構造で圧縮保存することである。

ﾂﾘｰ構造は、挿入順序及びｵﾌﾞｼﾞｪｸﾄ分布とほとんど関係なく、ﾊﾞﾗﾝｽのよいﾂﾘｰを生成できる BDﾂﾘｰ[7]を選択した。

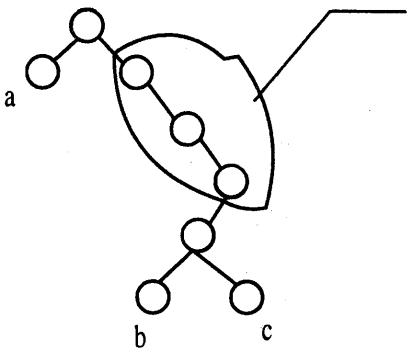
BDﾂﾘｰはある種類のﾊﾞｲﾅﾘ分割ﾂﾘｰ構造である。つまり、分割するたびに八分木のように空間の三方向全部に分割することではなく、一度に一つの方向だけ分割して、無駄に生成するﾉｰﾄを最小限に押さえられるﾃﾞｰﾀ構造である。

普通の二分木の違っているところは、BDﾂﾘｰは二分木のﾊﾞﾗﾝｽを破壊する内部ﾉｰﾄﾞ（一つの子ﾉｰﾄしか持っていないﾉｰﾄ）を削除することである。

たとえば、次の二次元の例で説明しよう。

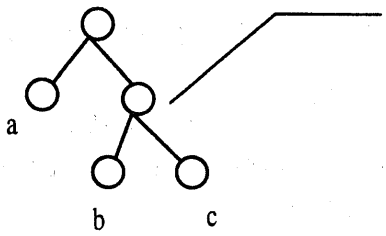


普通の二分木の場合



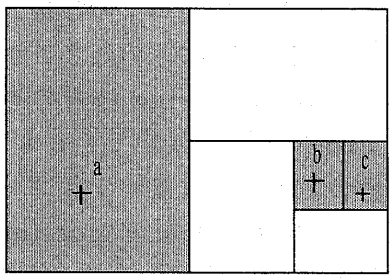
一つの子ノードしかない。ツリーのバランスを破壊して、ツリーのレベルを伸ばして、検索時間を長くしてしまった。
これらのノードの役は分割レベルを保存するだけで、別の方法で分割レベルが表せれば、削除できる。

BD木の場合



各ノードは自分が占めている領域を覚えて、無駄のノードが削除された。

表している領域は



5.4 Force Field Renderingの過程

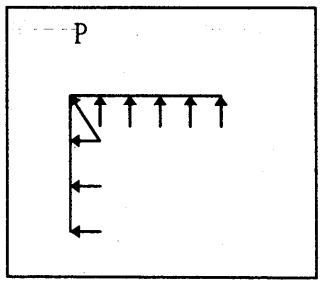
このようにして、BDﾂﾘｰは固定分割 Scheme(いつも空間の一番長い次元の真ん中から分割ｰｰﾂﾘｰ構造とｵﾌﾞｼﾞｪｸﾄの挿入順序の無関化)と無駄なﾉｰﾄﾞの削除のおかげで、ﾂﾘｰ構造の安定とﾊﾞﾗﾝｽの両立を達成した。

現在開発しているｿﾌﾄｳｴｱでは簡略な力学ﾓﾃﾞﾙを用いている。

$$F = -kx \tag{1}$$

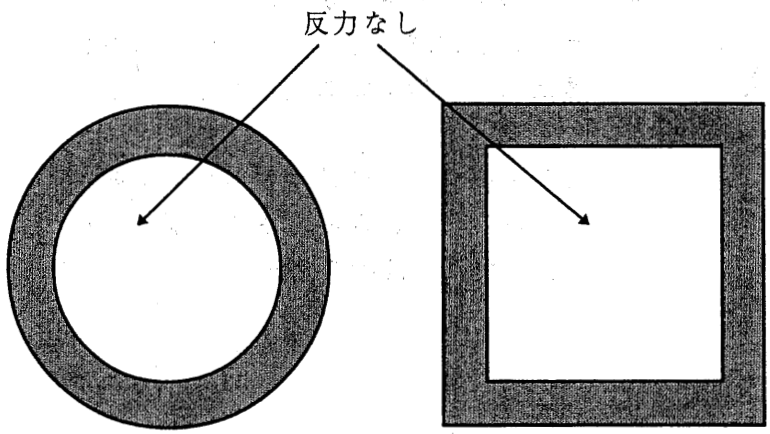
式(1)は仮想ｵﾌﾞｼﾞｪｸﾄを触るときに適用する。k はｵﾌﾞｼﾞｪｸﾄの剛性をあらわし、x はｵﾌﾞｼﾞｪｸﾄへの指先の侵入深さを表わす。

各ｵﾌﾞｼﾞｪｸﾄ毎に各ｻﾝﾌﾟﾘﾝｸﾞ点の反力を計算して、0ではないとき、BDﾂﾘｰに挿入する。二つのｵﾌﾞｼﾞｪｸﾄは同じ座標点を占める可能性がある。そういう場合には、BDﾂﾘｰ内部でﾍﾞｸﾄﾙ足し算で合成する。たとえば、ｺｰﾅｰの点Pの反力方向は水平、鉛直ではなく、斜線方向へ向いている。



Box Cone Cylinder Sphere 等のﾌﾟﾘﾐﾃﾌﾞは、境界からの一定の深

さまでしか反力が与えられない。これは、IndexFaceSet 等のﾎﾟﾘｺﾞﾝｵﾌﾞｼﾞｪｸﾄとの一致及びBDﾂﾘｰへの挿入量の減少の二重ﾒﾘｯﾄがある。[i]



6. まとめ

本研究ではVRMLﾌｧｲﾙでの形状を指先で感じることができる反力を有するﾌﾞﾗｳｻﾞを試作した。反力により視覚のみのときより現実感は大きく向上した。

Force Field Rendering 技術の導入により、不可能ではないかと思った複雑なﾜｰﾙﾄﾞも力覚装置PHANToMで体験できるようになった。ﾈｯﾄﾜｰｸを通じて構築されるVRMLﾌｧｲﾙを力覚で体験できることは豊かな仮想体験をもたらすものと確信できた。

これから、Force Field Rendering 技術を運動する物体への対応、ﾊｰﾄﾞｳｪｱ三次元ｱｸｾﾗﾚｰﾀの活用(Zﾊﾞｯﾌｧ等)、画像圧縮(DCT 等)によるﾒﾓﾘ消費量の軽減など方向へ発展するつもりである。VRMLﾌﾞﾗｳｻﾞのほうも、ﾊﾞｰﾁｬﾙﾍﾟﾝﾄ等の有用な機能を追加したい。

なお、本研究の一部は文部省科学研究費重点領域研究「人工現実感」、IBMｱｼﾞｱ奨学金による。

参考文献

[1]The Virtual Reality Modeling Language Specification
http://vrml.sgi.com/moving-worlds/spec/index.html

[2]VRML2.0 Final Draftｻﾝﾌﾟﾙﾊﾟｰｻｰ, ftp://vrml.sgi.com/pub/vrml2parser.zip

[3]VRMLScript Parser and Implementation,
http://vrml.sgi.com/moving-worlds/spec/vrmlscript.zip

[4]Alexander Stepanov and Meng Lee, The Standard Template Library (ANSI/ISO document)

[5]PHANTom Haptic Interface Installation & Technical Manual, SensAble DEVICES

[6]J.R.Levine, T.Mason, D.Brown 著, 村上列訳, lex & yaccﾌﾟﾛｸﾞﾗﾐﾝｸﾞ, ｱｽｷｰ出版局

[7]坂内正夫、大沢裕共著, 画像ﾃﾞｰﾀﾍﾞｰｽ, (株)昭晃堂

---

[i] もう一つ注意しなければならないことは、ﾎﾟﾘｺﾞﾝは一つの面しかない。つまり、ﾎﾟﾘｺﾞﾝの法線は内部から外部まで定義すれば、ﾎﾟﾘｺﾞﾝの内部は一定の深さまで反力があるが、外部はまったくない。はじめは変だと思う人もいるかも知れないが、これは当然であろう。
VRMLの仕様は確かに二面のﾎﾟﾘｺﾞﾝも許しているが、ﾊﾟﾌｫｰﾏﾝｽのためにDirect3Dのように強制的にbackface-cullingを有効し、つまり特別な手段を使わないとﾎﾟﾘｺﾞﾝの外面しかみえないﾚﾝﾀﾞﾘﾝｸﾞﾗｲﾌﾞﾗﾘの普及で、一つの面のﾎﾟﾘｺﾞﾝだけ使うVRMLﾌｧｲﾙが主流になり、ﾎﾟﾘｺﾞﾝの内部しか体感できないことは問題にならないと考えても大きな間違いではないたろう。