

クラスを用いた仮想環境の構成法

非会員 柳田 康幸 (東大)
非会員 舘 暲 (東大)

Virtual Environment Construction Method Using Class Library

Yasuyuki Yanagida, Non-Member and Susumu Tachi, Non-Member (The University of Tokyo)

To obtain highly realistic sensation of presence in virtual environment, natural three dimensional sensation of presence, real-time interaction between the operator and the virtual environment, and self-projection to the virtual environment are required. To implement these requirements, we have proposed the concept of "virtual human", which exists in the virtual environment and acts and feels as the operator in the real world would do. However, this concept has not been realized because of lack of systematic interface to access virtual environment data. In this paper, an object oriented method for constructing virtual environment using class library of the C++ language is proposed, to implement the concept of virtual human. This method was implemented on the system by human motion measurement and visual display, in order to show that the virtual human can be implemented using the proposed method.

キーワード: 人工現実感, テレイグジスタンス, オブジェクト指向, クラスライブラリ

1. まえがき

テレイグジスタンス^{(1)~(5)}は、人間が時間や空間の制約を超えて、時間もしくは空間あるいはその両方を隔てた環境に存在することを目指す概念である。近年注目されている人工現実感^{(6)~(9)}は、人間が現在存在している環境以外の仮想環境を、あたかもそこに存在しているような感覚を有して体験し、その環境で行動することを目的とする概念であり、テレイグジスタンスと人工現実感とは本質的に同一の概念である。テレイグジスタンスには、遠隔制御ロボットを利用した、人間が現在存在している世界とは別の実世界へのテレイグジスタンスと、コンピュータ等が創製した、実際には存在しないが極めて現実感のある仮想世界へのテレイグジスタンスが存在し、後者は狭義の人工現実感と呼ばれる。

仮想環境における高度な臨場感を実現するには、以下の条件を満たすことが重要である。

- 1) 仮想環境情報が自然な3次元空間を構成してオペレータに提示されること。
- 2) オペレータが仮想環境の中で自由に行動でき、オペレータと仮想環境との相互作用が実時間で行われること。
- 3) オペレータ自身が仮想環境の中に入り込んだような感覚が与えられること、すなわち、オペレータの仮想環境への自己投射性が実現されていること。

近年コンピュータ技術の進歩により、コンピュータグラフィックスの画像を実時間で更新し提示することが可能に

なり、人間運動の計測技術および3次元視覚提示技術とあいまって、人工現実感の概念を実現するシステムが出現し、活発な研究が行なわれている^{(6)~(9)}。しかしながら、高度な臨場感を有して仮想環境を介する作業を行なうにあたっては、従来の人工現実感システムでは次のような意味で十分とは言えない。

第1に、仮想環境に「カメラ」のみ、もしくはそれに加えて「手先」だけが配置されており、オペレータ自身のコピーを仮想環境に存在させるという概念がない。このため、「目」と「手先」が仮想空間中に浮遊しているような感覚があり、自分自身の体が存在しているという感覚、すなわち自己投射性に乏しい。遠景の中を動き回るだけの場合やあらかじめ決められた作業を実行する場合には自己投射性が乏しくても不自由はしないが、人間の高度な認識能力と作業能力が必要な不定型の作業を行なうには、この自己投射性が重要な要素となる。

第2に、人間の頭部運動のうち回転3自由度のみを利用し、並進運動情報を仮想環境に取り入れていないものがほとんどであり、体幹運動による頭部位置変化の再現が不可能である。また、上肢運動についても、手先の位置および姿勢のみの計測を行なっており、肘の位置など、人間の腕の冗長な運動を再現するのに十分な運動情報を計測しているものは少ない。このため、オペレータの運動が仮想環境での動きに十分に反映されず、これに基づいて視覚などの提示情報を生成するので、運動感覚と他の提示感覚情報と

の対応が十分でなく、仮想環境における自然な動きを実現しているとは言えない。

上記の要求を満たすため、著者らは「仮想人間」の概念を提唱している⁽⁵⁾。すなわち、実環境に存在するオペレータのコピーとしての仮想人間を仮想環境中に存在させ、実環境のオペレータの運動を反映した運動を行ない、仮想環境中を行動する。仮想人間は仮想感覚器（視覚、聴覚、触覚・力感覚など）を備えており、それら感覚器が検出した感覚情報をオペレータの感覚器に提示する。これは、実環境においてオペレータの分身として行動する遠隔制御ロボットを用いるテレグジスタンスとパラレルな概念であり、両者の相違は「テレグジスト」する先が実環境のロボットか仮想環境の仮想人間かというだけである。

ところで、この仮想人間の概念は、提唱されているものの、現在まで具体的には実現されていなかった。その理由として、以下のようなことが考えられる。現在までの人工現実感アプリケーションでは形状データを市販のCADによって作成しているものが多い。3次元形状データを対話的に作成するためにはこの方法は有効な方法であるが、このように外部的にデータの管理を行なっていたのでは実時間でオペレータとの相互作用を行なう場合にはその取り扱いが難しい。また、視覚の他に聴覚・力感覚等を提示したり、仮想環境に物理法則などを組み込む場合、このような方式では形状データだけで閉じているので、ユーザーが形状データとは別に独自に管理を行なう必要があり、複数感覚情報を系統的に発生させたり仮想環境の時間的な更新を行なうことは困難である。

そこで、仮想人間の概念を取り込んだ仮想世界を実現するには、上記のような既存ツールの組合せではなく最初から仮想環境構築を指向した、拡張性の高いツールが求められる。仮想環境構築ツールと呼ばれるものも出現しているが、そのプログラミングインタフェースは関数ライブラリの形で提供されているものがほとんどで、決められた関数群の中にユーザが新しい機能を組み込むことは困難であり、新しい試験研究的な要素を仮想環境モデリングに付加するためにはあまり役に立たない。

本論文では、仮想人間の概念に基づいて仮想環境に存在する仮想オペレータもしくは仮想遠隔制御ロボットへのテレグジスタンスを実現するため、実時間性を重視したオブジェクト指向に基づく仮想環境構成法の提案を行う。本手法はC++言語のクラスを利用しており、クラスライブラリの形で実装される。本クラスライブラリは、階層的記述が可能で物体に対する操作の一貫したインタフェースを備え、拡張性に優れている。また、本手法を人間運動の計測と視覚提示を行うシステム上に実装して、本手法による仮想環境構築の有用性を示す。

2. 仮想環境構成法の設計

(2.1) 仮想人間介入型人工現実感システムの必要条件
仮想人間の概念を取り入れた仮想環境の構成に必要な条件

を、以下に示す。

- 自然な3次元仮想空間が構成されること。仮想空間の物体の大きさや位置が、物理単位を使用して実寸で記述され、適切な位置関係で配置されていることが重要である。
- 仮想環境中の仮想人間が自然な運動を行なえること、すなわち仮想人間の運動の自由度が大きいこと。実オペレータの運動の計測の自由度を大きくとり、その運動を仮想環境中の仮想人間の運動に反映させることが必要である。自由度が制約されると、実環境でのオペレータの運動自由度が仮想人間の運動に反映されず、オペレータの運動に対応した感覚情報を生成することが不可能になる。頭部運動に関しては、回転3自由度だけでなく並進3自由度も計測することにより、上体の運動感覚が実現される。
- 仮想人間の運動に対応して、仮想世界の状態が更新されること。これにより、オペレータと仮想環境との相互作用が実現される。
- 高速であること。オペレータの運動が仮想人間の運動にリアルタイムに反映され、オペレータと仮想環境との相互作用が実時間で生じるために必要である。これには、提示情報の更新レートと、オペレータの運動の発生から更新された感覚情報が提示されるまでの時間遅れという2つの要素が存在する。更新レートが提示情報の「質」に影響することはもちろんであるが、時間遅れが存在すると、オペレータの運動感覚と生成される提示情報による感覚（視覚など）の不整合が生じ、酔いに似た極めて不自然な感覚が生じる。運動計測から感覚情報提示までを最小の時間遅れで行うことが重要である。これら2つの要素を満たすには、運動計測、仮想世界の状態更新、感覚情報生成のすべてを実時間性を考慮して設計し構成することが必要である。
- 環境への自己投射性があること。自分の体や腕などが見え、オペレータの運動に従って適切な位置に提示されることにより、オペレータが仮想環境の中へ入り込んだような感覚を与える。
- 仮想人間のサイズが実オペレータのそれを正確に反映していること。対象世界が人間の生活している世界と同等のスケールを有する場合には、環境への効果器である手やその位置を決定する腕のサイズ、感覚器である目の位置が実オペレータのものと一致していることが重要である。この条件が崩された場合、例えば眼間距離が実オペレータのものと異なると、輻輳角が異なるものになり、自然な距離感覚の発生が損なわれる。マイクロテレオペレーションなど、対象のスケールが異なっており、仮想人間と実オペレータのサイズを一致させることがそもそも不可能な場合には、仮想人間のスケールリングを注意深く行なう必要がある。
- 仮想人間の感覚器がとらえた感覚情報が、オペレータに正しく提示されること。視覚情報に関しては、視覚

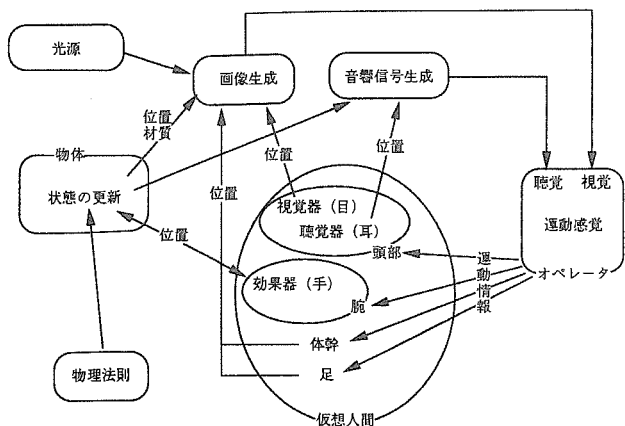


図 1 仮想人間介在型人工現実感システム

Fig. 1. Virtual reality system with virtual human.

提示デバイスの構造，設計パラメータを考慮した上で（視野角を一致させるなどして）画像生成を行なう必要がある。仮想人間の運動がオペレータの運動を正しく反映していて，この提示条件が守られることにより，オペレータの運動感覚と提示感覚情報との一貫性が保たれる。

- オペレータに提示を行なう感覚情報を生成する感覚チャンネルの定義や世界を支配する物理法則などの定義が行なえること。高度な臨場感をオペレータに与えるには，複数の感覚チャンネルによる提示が必要である。このため，視覚提示に限定しない汎用性，拡張性を有することが求められる。

上記の要件に基づき，運動感覚，視覚，聴覚のチャンネルを備え仮想人間が存在するシステムの概念を図1に示す。運動感覚は実オペレータの内部的な感覚であるが，運動情報を計測して仮想人間の運動に反映させることにより，仮想人間の感覚に反映する。物体の運動は，世界に組み込まれた物理法則と物体相互の位置関係，および仮想人間の効果器との位置関係により定められる。このようにして仮想人間も含めた仮想世界の状態が定められた上で，感覚情報の生成を行なう。視覚情報は両眼立体視を行なうため左右両眼用の画像を生成するが，仮想人間の目の位置で透視変換を行なった画像を生成する。仮想人間の目の位置が実オペレータの目の位置に追従しているため，実オペレータの運動に対応した視覚情報の生成が行なわれる。

(2.2) オブジェクト指向の導入 仮想環境中にはさまざまな物体が存在し，それぞれが位置，姿勢，色，形状などの属性を持っているが，これらのデータの管理をユーザが独自に行なうのは非常に複雑かつ見通しの悪い作業になる。C言語などのいわゆる手続き型言語ではデータとそのアクセス方法を管理づけることはユーザが自分の責任において行なう必要があり，仮想環境の統一的な記述を行なうのは困難である。C言語ではデータをひとまとめにして扱う方法として構造体が提供されているが，このメンバ

へのアクセスはスコープ範囲内で無制限に行なうことができる。これは一見便利なようであるが，同時に無秩序性を生み出す原因にもなりうる。すなわち，構造体の中のいくつかのデータが互いに関連していて一貫性の保持が必要とされるような場合，無制限なアクセスが許されているとユーザの不注意によりこれが崩される可能性がある。

以上のようなことから，仮想環境に存在する物体に関するデータはそれぞれの物体ごとに管理され，かつそれに対する操作のインターフェースが決められ整理されていることが要求される。これを行なうには，オブジェクト指向の概念を導入することが有効であると考えられる。オブジェクト指向を適用したモデル構成法の例としては，Lispを利用したもの⁽¹⁰⁾がある。これは強力な形状記述能力を持ち拡張性にも富む優れたものであるが，実時間で人間オペレータと仮想環境の相互作用を意図しているものではない。本論文で提案する仮想環境構成法は，人工現実感の必須要素である実時間での相互作用を重視する。

3. 仮想環境構成法の実装

前章の議論に基づき，仮想環境構成法の実装を行なった。オブジェクト指向言語はいくつかの種類が存在するが，本方式では，柔軟性，効率，汎用性などの観点から，C++言語のクラスを適用した。

(3.1) 実装した仮想環境構成法の特徴 提案する仮想環境構成法は，クラスライブラリに用意されている基本的なクラスを継承し，ユーザが物体定義クラスを記述していくという形をとるものである。本方式は，形状データを用意しておいて関数呼び出しにより仮想環境を構成していく方式と異なり，物体定義と物体を扱うインターフェースを統一的に記述していく点に特色を有する。

実装したモデル構成法的主要な特徴を，以下に示す。

- 仮想環境中の任意のオブジェクトへテレレイグジスト可能である。通常は，仮想環境に存在するオペレータ自身の「分身」である仮想オペレータ，あるいは遠隔ロボットのコピーたる仮想ロボットへのテレレイグジスタンスが行なわれる。
- 仮想環境記述のための基本的なクラスを用意しているため，特別な記述言語を用いなくても簡単に仮想環境を構築することができ，かつ柔軟な対応が可能である。
- オブジェクト指向の適用により，仮想環境中に存在する物体が持つ属性に関するデータの管理とインターフェースが整理されている。
- 物体の位置および姿勢は，物理単位に基づいて実寸で記述する。物体に対してグローバル/ローカルな位置・姿勢の指定や変更が可能である。
- 仮想環境中の物体は，物体プリミティブの組合せで表現する。
- 階層的に仮想環境を構築でき，複数の物体をブロック単位で扱うことが可能である。それぞれの物体は仮想世界を構成するツリーのノード情報を保持している。

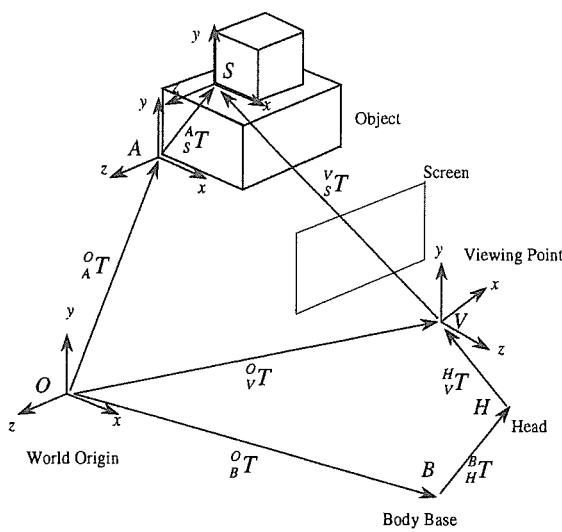


図2 座標系の記述

Fig. 2. Description of the coordinate system.

- 画像生成のみをターゲットとし、関数ライブラリで提供されているような従来の仮想環境構築ツールと異なり、自然な形で仮想環境物体の振舞いや属性の追加が行なえる。すなわち、用意されているクラスを継承して新しいクラスを定義し、そこに新しい属性や振舞いを表すクラスメンバを付加することにより、新たな感覚情報チャンネルの付加や物理法則の組み込みなどの拡張が、無理なく行われる。

(3.2) 仮想環境物体の記述方法 物体の位置および姿勢の指定は、その物体に固定された座標系（フレーム）が、基準となる座標系をどのように回転・平行移動して得られるかによって行なう。これには、 4×4 の同次変換行列を用いる。物体の形状は、その物体に固定されたフレームに関して記述された、物体上の点（ 4×1 ベクトル）の集合により表現される。

図2のように、以下の座標系を定義する。

- {O} : 世界基準座標系
- {B} : オペレータの体位置
- {H} : オペレータ頭部中心位置
- {V} : オペレータの視点（前方向き）
- {A} : 物体 A に付随するフレーム
- {S} : A の部分物体 S に付随するフレーム

以下、フレーム {X} から {Y} への変換行列を一般に ${}^X_Y T$ と表す。{A} に関してローカルに記述された物体 A 上の点 P_A を ${}^A P_A$ と書くと、世界基準座標系 {O} に関する点 P_A の表現は

$${}^O P_A = {}^O_A T {}^A P_A \dots\dots\dots(1)$$

である。次に、部分構成要素 S 上の点 P_S を {S} に関して ${}^S P_S$ として記述すると、 P_S の {A} に関する記述は

$${}^A P_S = {}^A_S T {}^S P_S \dots\dots\dots(2)$$

であるから、世界基準座標系に関しては

$${}^O P_S = {}^O_A T {}^A P_S = {}^O_A T {}^A_S T {}^S P_S = {}^O_S T {}^S P_S \dots\dots\dots(3)$$

と表現される。物体全体の位置や姿勢 (${}^O_A T$) が変化しても、部分要素 S は ${}^A_S T$ というローカルな変換を保持しておけば、正しい世界座標の中の位置が計算される。

仮想環境の物体上の点は、モデリング変換（仮想空間の中での物体の移動・回転）、ビューイング変換（視点の位置・方向の指定）、投影変換（平面への投影、透視変換を用いる）を施してウィンドウ上に表示される。このとき、透視変換のパラメータである視野角を使用するデバイスの視野角と一致させ、視点を仮想人間の左・右眼の位置にすることが重要である。

ビューイング変換には、オペレータの体の仮想空間内での位置・姿勢、体と頭部の位置・姿勢の関係、頭部の中心と左眼/右眼の位置関係が全て含まれる。すると、世界基準座標系からオペレータの視点フレームへの変換行列は

$${}^O_V T = {}^O_B T {}^B_H T {}^H_V T \dots\dots\dots(4)$$

となる。頭部運動の計測値は、 ${}^B_H T$ に反映される。

次に、視点 {V} から物体 {S} を見ることを考える。世界基準座標系で見た物体 {S} の位置・姿勢は ${}^O_S T$ と表記され、これはモデリング変換に該当する。 ${}^O_S T$ を視点フレームを経由して表現すると

$${}^O_S T = {}^O_V T {}^V_S T \dots\dots\dots(5)$$

と表されるので、視点フレームを基準とした物体 S の変換行列は

$${}^V_S T = ({}^O_V T)^{-1} {}^O_S T \dots\dots\dots(6)$$

となり、物体フレーム {S} の上でローカルに記述された点の座標にこの変換行列を掛けてさらに透視変換を施すことにより、左/右眼用の画像が得られる。

(3.3) 描画処理効率化のための手法 実時間で視覚情報提示を行なうためには、高速な描画処理は必須である。本節では、描画処理の高速化を行なうために使用した手法について述べる。

(1) 幾何コンパイル (3.2)節のように、それぞれの物体でその基準となる座標系からの相対的な位置および姿勢を表す同次変換行列を保持し、世界を形作るツリーの根元から順番にそれぞれの変換行列をかけてやればその物体のグローバルな位置および姿勢が計算できるが、固定された物体、あるいはブロックの中で位置関係が変わらないような要素の場合には、毎回これを計算することは効率が悪い。そこで、このような物体についてはあらかじめ世界原点からの絶対位置を計算しておくことにする。これを、ここでは幾何コンパイルと呼ぶことにする。表示物体の数が多き場合には、幾何コンパイルによる計算省略の効果は大きく、画像表示の高速化につながる。

幾何コンパイルを行なうには、まずその物体から世界構成ツリーを根元に向かってたどりながら変換行列の計算を

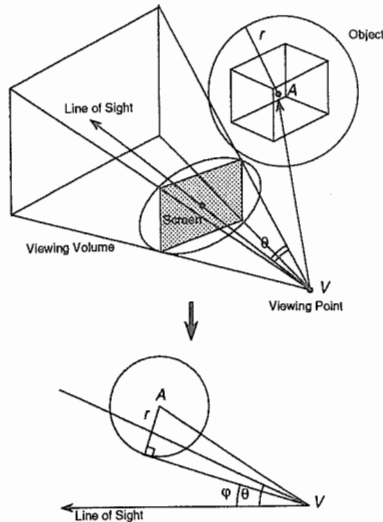


図3 ブロッククリッピング
Fig. 3. Block clipping of objects.

行なう。次に、その物体上の頂点座標にこの変換行列をかけることにより、物体上の点を幾何コンパイル原点フレームで表した座標値が計算される。

幾何コンパイル原点は、世界基準座標系にとることも、任意の物体にとることも可能である。固定された物体の場合は幾何コンパイル原点を世界基準座標系にとり、ロボットアームのリンクが複数のオブジェクトで構成される場合などは、幾何コンパイル原点をそのリンクに付随したフレームにとればよい。

幾何コンパイルされている場合とされていない場合との相違はその都度幾何コンパイル原点フレームからその物体までの変換行列の計算を行なうかどうかだけであり、全く区別せずに利用可能である。

(2) ブロッククリッピング 画像生成を高速化するため、視点の位置および方向、物体の位置およびおよその大きさを元に、明らかに視野に入らない物体はブロック単位で描画対象から除外する機能を備えている。図3において、物体を内包する半径 r の球を考え、視野の中心方向と物体中心方向が作る平面上において、視点を通る球の接線と視野中心方向がなす角を ϕ 、視野の中心から視野の隅までのなす角を θ とする。 $\theta < \phi$ ならば物体の描画は行なわれない。ウィンドウのクリッピングはグラフィックスワークステーションの機能によっても行なうが、これはあらかじめ描画を行なう物体の数を減らしておくためのものである。

(3.4) クラスライブラリ 本仮想環境構成法の基盤となるのは、仮想環境物体を定義するためのクラスライブラリである。図4に、開発したクラスライブラリの構造を示す。クラスライブラリには、以下のようなクラスが存在する。

- **Coord3**: 3次元座標および3次元ベクトルの記述と取り扱いを容易にするために定義している。このクラスに属する3次元座標/ベクトルの初期化、設定、スカラ

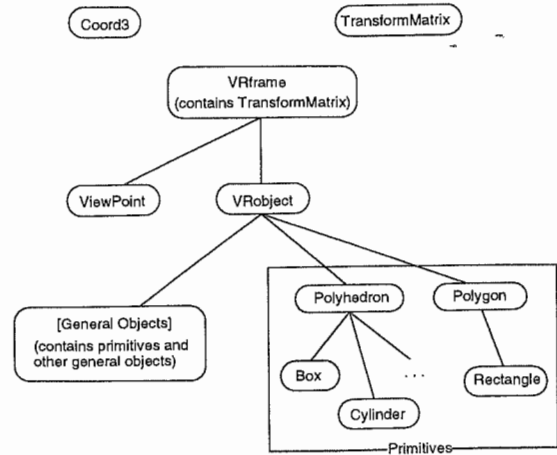


図4 クラスライブラリの構成
Fig. 4. Configuration of the class library.

倍、絶対値、および3次元座標/ベクトル同士の加減算を定義しており、加減算およびスカラ倍については、演算子(+, -, *, /=)の多重定義によりC言語の組み込み変数(int, floatなど)と同様の記述が可能である。

- **TransformMatrix**: 同次変換行列(4×4)を表すクラスである。初期化、位置の設定および値の取得、平行移動、スケーリング、軸の回りの回転、逆行列、同次変換行列同士の乗算を定義している。また、3次元座標との乗算(回転+平行移動)、同次変換行列の回転行列部分によるベクトルの回転を定義している。
- **VRframe**: 「フレーム」を表すクラスである。このクラスのインスタンスには座標変換行列が付随しており、基準フレームに対するそのフレームの座標変換を保持している。このクラスのインスタンスに対しては、行列全体の設定、位置指定、位置の獲得、平行移動、スケーリング、軸の回りの回転、逆変換が行える。
- **VRObjct**: 「汎用仮想オブジェクト」を表すクラスであり、「フレーム」クラスを継承している。このクラスには、フレームクラスから継承した座標変換行列が存在する。この変換行列は、その物体の「親」物体に対する、その物体の相対的な(ローカルな)座標変換を保持し、物体のローカルな並行移動、回転などはこの変換行列に対して行なわれる。加えて、幾何コンパイル(3.3節)のために、幾何コンパイル原点に対する位置および姿勢を保持するための座標変換行列が用意される。また、世界の木構造を保持するノード情報(親物体へのポインタと子物体へのポインタのリスト)があり、物体間の関係を記述している(図5)。さらに、画像生成用に物体の材質を表すクラスへのポインタがあり、描画処理に使用される。
- **ViewPoint**: 「視点」を表すクラスであり、「フレーム」クラスを継承して定義している。仮想環境中に存在する物体の位置と姿勢をこの視点クラスの属性に指定することにより、その物体へのトレイグジストが行な

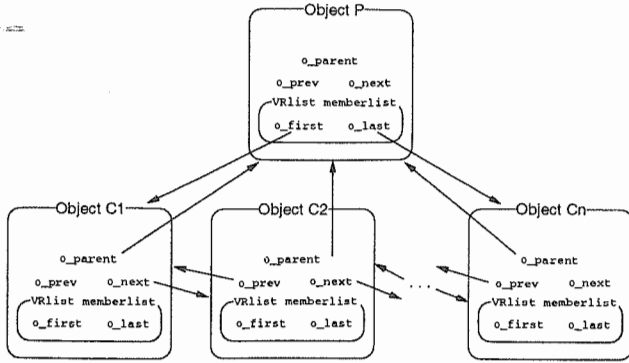


図5 オブジェクトの階層構造
Fig. 5. Object Tree.

れる。

- 物体プリミティブ: 「汎用仮想オブジェクト」クラスを継承して、基本的な物体(プリミティブ)を用意している。汎用プリミティブのためのクラスとして多面体(Polyhedron)とポリゴン(Polygon)があり、これらを継承してそれぞれのプリミティブクラスが作成される。現在、この物体プリミティブとしては直方体および円柱といった単純なものが用意されているが、これらの単純なプリミティブを組み合わせたユーザーがプリミティブを定義したりすることにより、複雑な形状の物体を表現することも可能である。

実際の物体クラスは、「汎用仮想オブジェクト」クラスを継承して定義する。これらのクラスの中に形状プリミティブを入れたり、別の物体クラスあるいはその物体クラスへのポインタをメンバにすることにより、階層的な世界の記述を行なう。物体の状態変更は、定義したクラスのメンバ関数を通して行なう。基本的な運動を記述したクラスを設計し、そのクラスを継承して具体的な物体を記述すれば、それらすべてに共通な運動法則を定義することが可能である。

仮想環境の描画は、視点位置をトレイジストする物体の位置に設定した上で、世界を構成するツリーの根元のオブジェクトに対して描画関数を呼ぶことにより行なわれる。世界構成ツリーをたどりながら子オブジェクトの描画が行なわれるので、ツリーの根元のオブジェクトに描画命令を送るだけでそのツリーに含まれるすべてのオブジェクトの描画が行なわれる。

4. 仮想環境構成実験システムの構築

本手法による仮想環境構成の有効性を示すため、人間の運動計測と視覚提示を行なうシステム⁽¹¹⁾を構築し、本手法により仮想環境の構成を行なった。

(4.1) システムのハードウェア構成 システムの構成の概要を図6に、システムのプロック図を図7に示す。

オペレータの運動は、人間運動計測装置によって実時間で計測され、計測用計算機に読み込まれる。その情報は仮想環境構成・画像生成用計算機に送られ、仮想環境モデル

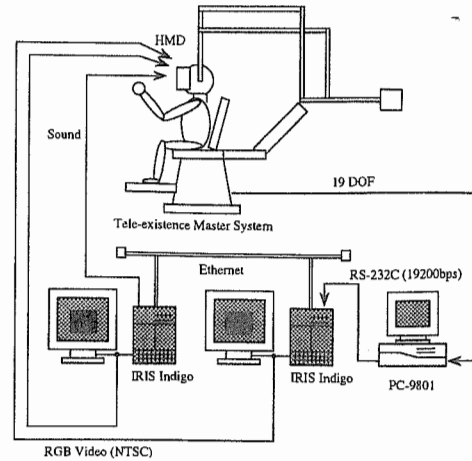


図6 システムの構成

Fig. 6. Hardware configuration of the system.

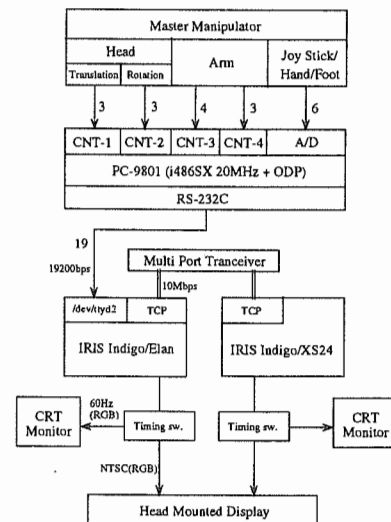


図7 システムのプロック図

Fig. 7. Block diagram of the system.

と人間運動の情報を用いて左眼および右眼用画像が生成される。この画像を頭部搭載型ディスプレイ(HMD)に表示し、オペレータに視覚提示を行う。

以下、各々の部分について説明を行う。

(1) 人間運動計測部 オペレータの運動の計測は、トレイジスタンス作業用マスタシステムによって行なう。この装置では、次のような運動計測を行なっている。

- a. 頭部運動: 並進3自由度, 回転3自由度。並行リンク機構に取り付けられたロータリーエンコーダで計測。
- b. 上肢運動: 7自由度。エグゾスケルトン型マニピュレータのロータリーエンコーダで計測。
- c. ハンド開閉: 1自由度。ポテンシオメータで計測。
- d. ジョイスティック: 3自由度。ポテンシオメータで計測。
- e. フットペダル: 左右1自由度ずつ, 計2自由度。ポテンシオメータで計測。

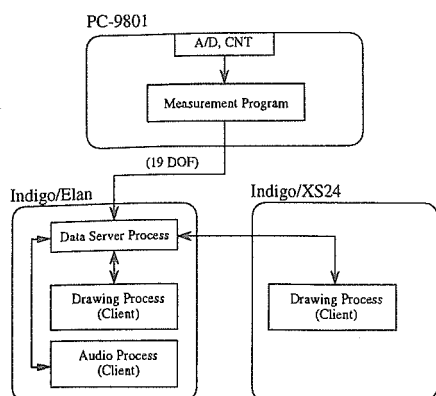


図 8 プロセス構成

Fig. 8. Process configuration.

以上合計 19 自由度の情報を計測している。これらの情報はカウンタボードおよび A/D 変換ボードを通して計測用計算機 (NEC 製 PC-9801BX, CPU i486SX 20MHz + ODP) に読み込まれる。本計測システムでは、市販されている 3 次元位置計測デバイス (磁気センサ) などと比べて、非常に高速に計測を行うことができる (更新レート 1kHz 以上, 時間遅れ 1ms 以下)。この大きな自由度と高速性により、仮想人間の自然な運動と実時間での仮想人間の状態更新が実現される。

(2) 仮想環境構成・画像生成部 Silicon Graphics 社のグラフィックスワークステーション IRIS Indigo/Elan および XS24 (CPU R3000/3010 33MHz) を Ethernet で接続して運用している。これらのマシンは高速・高機能なグラフィックスハードウェアを搭載し (Elan: 100k 四角形ポリゴン/秒, XS24: 25k 四角形ポリゴン/秒), 仮想環境の画像をリアルタイムで生成することが可能である。また, DSP を含むオーディオサブシステムを搭載しており, 視覚・聴覚情報を統合的に扱うことに適している。

(3) データの流れ 計測用計算機 (PC) から仮想環境構成・視聴覚情報生成用計算機へは, シリアル回線 (19200 bps) を用いて運動情報の転送を行っている。計測される 19 自由度の情報を全部送る場合, データの更新レートは毎秒 20 回程度である。

図 8 に示すように, PC からシリアル回線を経由して送られてきたデータは, 一旦ワークステーション上のデータサーバプロセスで受け取られる。データサーバプロセスと描画プロセスは同一の世界モデルを持っており, 仮想世界の更新はサーバプロセスで一元的に実時間で計算される。更新された情報は UNIX のプロセス間通信 (socket) の機能を利用して描画プロセスなどのクライアントへ送られる。これにより, ネットワーク上に分散された描画プロセス間で矛盾のない画像提示を行なうことができ, 処理の負担の大きい左眼, 右眼用の画像生成を別々のマシンで実行させることが可能となる。

(4) 画像の提示 本システムでは, ワークステーションのフレームバッファ上に左眼/右眼用の画像を展開し, それぞれを NTSC タイミングの RGB 信号で出力して頭部搭載型ディスプレイ (HMD) に提示している。HMD は人間運動計測装置の頭部運動計測部に取り付けられたヘルメットに固定されており, 左右両眼独立に画像を提示する。HMD は左右両眼用にそれぞれ 720 × 240 画素のカラー LCD を用いており, 提示距離 1m の位置に視野角 40° で提示されるよう設計されている。

(4.2) 仮想環境構成例 構築したシステムに提案する仮想環境構成法を実装して, テレイグジスタンス型マスタ・スレーブシステムを仮想環境中に構築した。環境はテレイグジスタンス実験用マスタシステムとスレーブロボットが設置されている実験室およびその建物の廊下で構成されている。

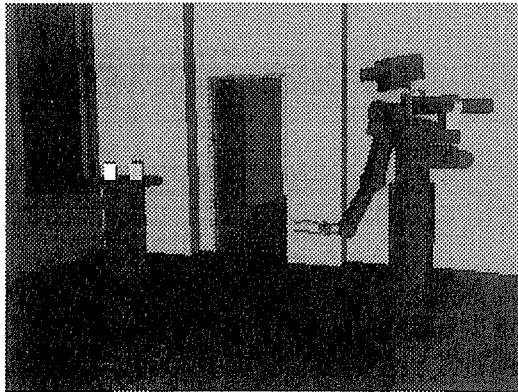
作成した仮想環境は, すべて実寸により記述されており, 実環境の物体の大きさや配置を正確に反映している。このため, 自然な 3 次元空間が構成されている。

本プログラムでは, 自分が仮想オペレータとして仮想環境内に存在するモードと, 仮想ロボットの中に入り込んでテレイグジスタンスによる作業を行うモードが存在する。すなわち, 仮想オペレータへのテレイグジスタンスと仮想ロボットへのテレイグジスタンスが行なえる。仮想オペレータの運動と仮想スレーブロボットの動きはテレイグジスタンスによる作業を想定して連動しており, スレーブロボットは仮想オペレータにより操作されるマスタシステムと同じ動作を行う。

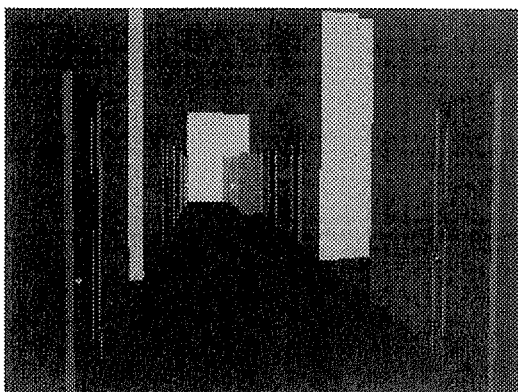
仮想オペレータモードの時は, 手を伸ばすと仮想人間の腕や手が見え, 下を向くと足や体が観察できる。この仮想人間は仮想マスタシステムのオペレータ席に座っており, 仮想マスタシステムのアームやジョイスティックが実オペレータの操作に応じて動き, 実際に操作を行っている感覚を与える。仮想ロボットモードの時は, 実環境でのテレイグジスタンス作業と同様に手を伸ばすとロボットのアームが見え, 仮想ロボットに乗り移ったかのような感覚を与える。これにより, 仮想環境への自己投射性が実現されている。

仮想オペレータと仮想スレーブロボットは, ジョイスティックの操作によって仮想環境内を自由に動き回ることができる。これは, オペレータが仮想空間内を自由に行動するという要件を実現したものである。

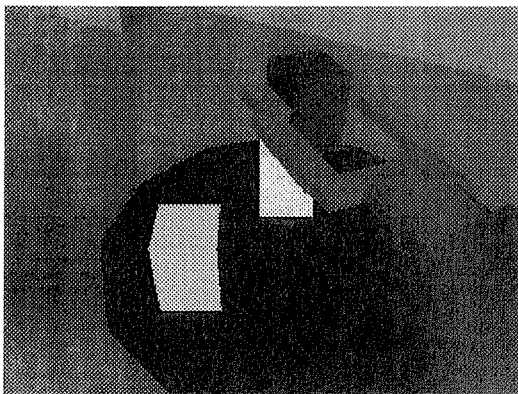
また, 仮想環境内に物理法則を適用して, 簡単な積み木作業と, 天井から吊したピンポン玉を打つ作業を仮想環境内に実現した。これはオペレータと仮想環境との相互作用の実現例である。これらの作業は, 仮想オペレータ自身が行なうことも, 仮想スレーブロボットにテレイグジストして行なうこともできる。この運動の法則の記述は, クラスのメンバ関数としてその物体自身が状態を更新するように記述されている。また, 簡単な複数感覚情報チャンネルの実現として, ピンポン玉が手と接触した時に音を発生し, オペレータに提示するようにした。仮想環境の画像の例を



(a) 仮想スレーブロボット
(a) Virtual slave robot.



(b) 建物の廊下
(b) Corridor of the building



(c) 仮想環境における積木作業
(c) Block building task in virtual environment.

図9 仮想環境の画像例

Fig. 9. Example images of the virtual environment.

図9に示す。

仮想環境中に存在する各物体は、次のように構成されている。

- マスタマニピュレータ：336 ポリゴン
- スレーブロボット：834 ポリゴン
- 廊下のドア1枚：122 ポリゴン
- 窓：30 ポリゴン

構築された仮想環境全体のポリゴン数は約5000である。表示対象ポリゴン数が多いのでこれらを無条件に描画すると実時間での制御は難しいが、(3.3)節で述べた幾何コンパイルおよびブロッククリッピングの利用と視点からの距離に応じた物体表示の簡略化により、描画処理の高速化が図られている。視点からの距離に応じた物体表示の簡略化は、物体自身が視点からの距離を検知し、表示ポリゴン数を調整している。

全19自由度を制御し、これらの物体を画像表示して、平均10数Hzの描画レートが得られている。(3.3)節の手法の効果について述べると、最もポリゴン数の多い建物のブロック化により2~3倍、さらに物体との距離による表示簡略化、ブロッククリッピング、幾何コンパイルにより25%から90%高速化されている。また、プログラムの実行時間については、計算機処理能力の大部分が描画処理に費やされており、動的な世界更新を行なうサーバプロセスの描画プロセスに対する負荷の比率は1/100程度、また描画プロセスにおける描画処理以外の部分の実行時間は数%程度に留まっている。

計測データの転送および画像生成のために若干の時間遅れが存在し、データ転送による遅れについては改善の余地が残されているが、高速な計測システムの効果と高速性を意識したプログラムにより磁気センサを用いたシステムなどと比べると時間遅れは非常に小さくなっており、運動感覚と視覚との良好な一致性を示している。

5. むすび

仮想人間へのテレイグジスタンスの概念を実現するため、オブジェクト指向に基づいてC++言語のクラスライブラリを用いた仮想環境構築法を提案し、人間の運動計測と視覚提示による高速な人工現実感システムを構成して、本システム上に提案する仮想環境構築法を用いた仮想環境の構築を行なった。その結果、本手法により仮想人間の概念が工学的に実現可能であることを示した。

(平成6年6月29日受付)

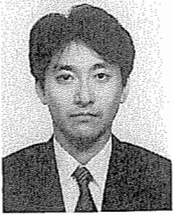
文 献

- (1) 舘 暉, 阿部 稔: 「テレイグジスタンスの研究 第1報—視覚ディスプレイの設計—」, 第21回計測自動制御学会学術講演会予稿集, 167~168 (昭57-7)
- (2) 舘 暉, 荒井裕彦: 「テレイグジスタンスにおける視覚情報提示系の設計と評価」, 日本ロボット学会誌, 7, 4, 314~326 (平1-8)
- (3) S.Tachi, H.Arai and T.Maeda: "Development of an Anthropomorphic Tele-Existence Slave Robot", Proceedings of the International Conference on Advanced Mechatronics (ICAM), 385~390 (1989)
- (4) 前田太郎, 荒井裕彦, 舘 暉: 「頭部運動追従型両眼視覚提示装置の設計と評価」, 日本ロボット学会誌, 10, 5, 655~665

(平 4-5)

- (5) 舘 暉, 前田太郎: 「人工現実感を有するテレイグジスタンスロボットシミュレータ」, 電子情報通信学会論文誌, **J75-D-II**, 2, 179~189 (平 4-2)
- (6) 岩田洋夫: 「大規模仮想空間を歩行するための人工現実感」, *Human Interface*, 5, 2, 49~52 (平 2-5)
- (7) M.Hirose et al: "A Study of Synthetic Visual Sensation through Artificial Reality", ヒューマン・インタフェース研究論文集, 1, 1, 19~26 (平 4-11)
- (8) 竹村治雄, 岸野文郎: 「人工現実感によるヒューマンインタフェース」, テレビジョン学会誌, **44**, 981~985 (平 2)
- (9) 佐藤 誠, 平田幸広, 河原田弘: 「3次元モデリングのための仮想作業空間」, 3D映像, 4, 2, 26~35 (平 2)
- (10) 松井俊浩, 稲葉雅幸: 「Euslisp: オブジェクト指向に基づくlispの実現と幾何モデラへの応用」, 情報処理学会記号処理研究会, number 2 in 50, (平 1)
- (11) 柳田康幸, 舘 暉: 「センサフュージョン機能を有する人工現実感システム (第1報)—システムの構成—」, 第10回日本ロボット学会学術講演会予稿集, 1087~1088 (平 4-10)

柳田 康幸 (非会員) 1964年10月17日生まれ。88年東京大学計数工学科卒業, 90年東京大学大学院工学系研究科修士過程修了。同年東京大学先端科学技術研究センター助手, 現在に至る。主として人工現実感システムの研究に従事。



舘 暉 (非会員) 1946年1月1日生まれ。68年東京大学工学部計数工学科卒業, 73年東京大学大学院工学系研究科博士過程修了。同年東京大学工学部助手。75年通産省機械技術研究所所員, その後主任研究官, 遠隔制御課長, バイオロボティクス課長。79年から80年, マサチューセッツ工科大学客員研究員。89年東京大学助教授, 92年東京大学教授, 現在に至る。

