

テレイグジスタンスにおける
視覚と運動感覚の整合性に関する研究

柳田 康幸

目次

1	序論	1
1.1	研究の背景	1
1.2	VR およびテレプレゼンスシステム概観	3
1.2.1	情報の流れとインタフェースレベル	3
1.2.2	VR システムの品質を決定する要素	6
1.3	運動計測と視覚提示によるシステム	8
1.3.1	運動計測デバイス	8
1.3.2	視覚提示デバイス	9
1.3.3	デバイス/サブシステム間接続	10
1.3.4	ソフトウェア	10
1.4	本論文の目的	10
1.5	本論文の構成	11
2	Virtual TELESAR の構築	13
2.1	システム全体の設計指針	13
2.2	ハードウェア構成	15
2.2.1	人間運動計測部	17
2.2.2	VR 空間管理・画像生成部	18
2.2.3	信号と情報の流れ	18
2.2.4	画像の提示	18
2.3	ソフトウェアの設計指針	19
2.4	VR 空間管理・画像生成用ライブラリ	21
2.4.1	VR 空間における物体の記述方法とシーングラフ	21
2.4.2	オブジェクト指向の導入	25
2.4.3	描画処理効率化のための手法	26
2.4.4	ライブラリの実装	30
2.5	アプリケーションプログラムの枠組み	35
2.5.1	分散処理とネットワーク対応	35
2.5.2	パラメータ制御	41
2.6	VR 空間のモデリング	44
2.7	まとめ	48
3	HMD 使用時の視覚系パラメータ整合性の影響	49
3.1	概要	49
3.2	視覚・運動パラメータの整合性に関する幾何学的解析	51
3.2.1	HMD のパラメータ	51

3.2.2	パラメータ不整合による空間の静的変形	52
3.2.3	頭部回転運動による動的変形と物体の移動	54
3.3	頭部回転運動の倍率制御による画像の安定化	57
3.3.1	安定化の原理	57
3.3.2	スクリーン上の点の流れに基づく導出	57
3.3.3	両眼立体視時の空間変形に基づく導出	60
3.4	評価実験	62
3.4.1	実験装置	62
3.4.2	実験 1: 並進パラメータに関する評価	62
3.4.3	実験 2: 回転運動に関する評価	64
3.5	まとめ	66
4	視覚提示デバイスの特性分析	68
4.1	視覚提示デバイスの種類	68
4.2	HMD と IPT の長所と短所	70
4.3	操作者の頭部運動に対する提示画像の挙動	72
4.4	正確な立体視のための透視変換	75
4.5	まとめ	78
5	HMD 利用時の頭部回転運動に対する画像安定化	79
5.1	概要	79
5.2	回転運動に関する HMD 提示画像の補正手法	80
5.2.1	基本原理	80
5.2.2	提示画像の遅延の分析	82
5.2.3	回転角補正量の算出法	82
5.2.4	実装手法の検討	83
5.3	画像歪みの見積	84
5.3.1	角度誤差による見積	86
5.3.2	画面上距離誤差による見積	88
5.4	実時間シミュレーションによる検証	89
5.5	試験的実装	90
5.5.1	システムの機能構成	90
5.5.2	プラットフォームの選択とソフトウェアの構造	91
5.5.3	課題と今後の展望	92
5.6	まとめ	92
6	固定スクリーンを利用した実環境へのトレイグジスタンス	93
6.1	概要	93
6.2	固定スクリーンをトレイグジスタンスに利用する際の問題点	93
6.2.1	形状の変化する非対称透視変換	93
6.2.2	頭部運動追従の必要性	94
6.2.3	カメラの光軸方向の問題	99
6.3	トレイグジスタンス視覚系の構築手法	100

6.4 カメラシステムの設計	105
6.4.1 姿勢保持機構の設計	105
6.4.2 提示用画像操作部の構築	107
6.5 まとめ	107
7 結論	108
謝辞	110
参考文献	111
関連発表	117

第 1 章

序論

1.1 研究の背景

テレグジスタンス (Teleexistence) は人間が実在する場所以外の遠隔環境, もしくはコンピュータの創成したバーチャル環境へ臨場感を持ちつつ存在し行動する概念であり, 前者を実環境へのテレグジスタンス, 後者をバーチャル環境へのテレグジスタンスという. 後者はまた, 狭義のバーチャルリアリティ (Virtual Reality: VR) と呼ばれる. テレグジスタンスの概念は 1980 年代初頭に館ら [58] によって提唱され, 日本においては通産省の極限作業ロボットプロジェクトなどを通してその工学的実現可能性が立証されてきた. 海外においては, ほぼ同時期にテレプレゼンス (Telepresence) [1] [13] という名称で発表され, 研究開発が進められた. これらは主としてロボティクス分野における活動であるが, 他にも臨場感通信, 計算機インタフェース, シミュレーションの視覚化, 設計支援, インタラクティブアートなどの分野で共通の目標を目指していることが認識され, 1990 年にそれらの分野の研究者が一堂に会してお互いの活動の共通性を認識した上で, バーチャルリアリティとしての共通性を意識しつつ各分野での研究開発と実用化が活発に進められている.

「バーチャルリアリティといえばヘッドマウントディスプレイ (Head Mounted Display: HMD) にデータグローブ」というスタイルが一般に認知されたのは 1990 年代前半であり, 以降典型的なバーチャルリアリティシステムの形態として定着している. HMD は 1965 年に Sutherland によって提唱された「究極のディスプレイ」[56] のコンセプトを実現する手段として開発され [57], 元来頭部運動を計測する装置を併用して装着者の運動に実時間で対応した画像を提示することにより, 周囲の空間を視覚的に代替するものである. このコンセプトと HMD は当時のコンピュータの処理能力や計測技術, 画像提示デバイスなどの制約もありすぐに広まることはなかったが, 1980 年代以降要素技術の発展により複数の研究機関において試作が行われ, 1980 年代末には商品として市販されるようになった. さらにその後デバイスの小型・軽量化が進み, 1990 年代後半になって一般消費者も入手可能な HMD が商品化されるに至り, HMD はもはや特別な施設や実験室の中の存在ではなくなった. ところが, これらの一般向け HMD は単なるパーソナルテレビとしての用途をターゲットとしており, 情報機械に対するインタフェースをバーチャル化し人間の視覚空間をインタラクティブにカバーするという様態が実現されている例は少数である. つまり, 人間の運動に実時間で対応したテレグジスタンス・VR システムは未だ実験室や少数のショールームなどの施設から外に出ているとは言い難い状況である.

この HMD の場合のように, 個々のデバイスとしての性能や品質は上がってきているにもかかわらず, インタラクティブなバーチャルリアリティ/テレグジスタンスシステムが一般社会に浸透していくスピードはそれほど速くない. これはすなわち, 従来の VR システムが一般消費者の, あるいは産業界の使用に耐えるレベル (合格ライン) に到達していないことを示唆している. 実際, 1990 年代初頭には既に数々の VR システムが構築されており, 上記の HMD を利用したシステムでは確かに

「右を向けば右方向の画像が表示される」のではあるが、どうも「そこに世界がある」ように感じられないのである。これには数々の問題が内包されており、もちろん、VR 世界を視覚的に表現するためのコンピュータグラフィックス (Computer Graphics: CG) 画像の精密さも問題である。が、筆者が違和感を覚えたのはそこではない。むしろ、そこにあるはずの物体や周囲の環境がそこにあるように感じられないことこそが違和感であった。この違和感を解消することはできないだろうかというのが本研究の発端である。

筆者がこのような問題意識を持つに至るには、そうした VR システムに対して比較対象のリファレンスとなるシステムの存在が重要な役割を果たした。それは、1980 年代を通して研究開発が進められた（実環境への）トレイグジスタンス実験システム [31] [58]–[62] の存在である。このトレイグジスタンス実験システムは機械式リンク機構により人間の頭部および上肢（右腕）運動を計測し、人間型ロボットの頭部と腕が操作者と同様の動きをするよう制御するとともに、ロボット頭部に搭載されたステレオカメラで得られるビデオ画像をリンク機構に取り付けられた HMD へ提示している。一見、コンピュータの創製するバーチャル世界を提示する狭義の VR システムの方が、機械装置の物理的な制御を伴わない分だけ人間の運動に対する振る舞いは有利なように思えるが、実際に当時のシステムを体験してみると、感触はむしろ逆であった。つまり、HMD を装着して周囲を見渡したとき、トレイグジスタンス実験システムの方がはるかに周囲の環境が「そこに存在する」ような感覚が得られたのである。

この体験が意味するところは何であろうか。一つには、実環境版トレイグジスタンスシステムが実写ビデオ画像を提示しているのに対し、狭義の VR システムでは VR 世界を CG 画像により提示しており、当然提示される世界の画像としてのリアルさが圧倒的に異なるという点が考えられる。コンピュータグラフィックスの世界では 1990 年代を通してフォトリアリスティックなグラフィックス画像生成の研究が盛んに行われ、映画のような用途で時間を十分かけられるオフライン CG においては実写映像と区別がつかないほどの品質を実現することが可能になってきているが、VR のように秒間数十枚もの画像を生成する必要があるリアルタイム CG の分野では、そこまでの品質を実現できていない。しかしながら、前述したように、筆者が従来の VR システムにおいて違和感を感じたのは、そうした提示画像自体の品質ではなく、自分が頭部を動かした場合に周囲の世界がそこに存在するように感じられなかったり、自分の腕を目の前で動かした場合に提示される腕や手が「自分のものである」という感覚が乏しいことである。このような違和感は、一方的に画像を提示される映画のような場合には存在し得ない問題であり、VR やトレイグジスタンスのように実時間で遠隔・VR 世界とのインタラクションが生じる場合に特有の問題であることは明らかである。

このように、本研究において着目するのは、VR のインタラクティブ性である。インタラクティブ性に対する要求レベルはどのようなアプリケーションを構築するかによって様々であり、言い換えると実時間性という言葉は、それが使用される局面によって意味合いが全く異なってくる。例えば、インターネットにおける Web においては、ユーザがリンクをクリックしてから「待てる許容範囲」（数秒～数十秒）で情報が表示されれば十分にインタラクティブであり、ビデオ・オン・デマンドにおいては送られてくる情報が途切れないという実時間性の要求が加わるものの、ユーザのリクエストに対して再生が始まるまでの応答時間は Web と似たようなものである。これに対し、VR は人間の感覚情報をすり替えるものであるから、感覚特性が実時間性レベルの可否を決めるポイントとなる。

こうしたインタラクティブシステムを構築するにあたっては、デバイス単体としての進歩もさることながら、同時にシステム全体を注意深く構成することが必要不可欠であり、まさにシステム工学の観点が要求される。つまり、「VR 用のデバイスをいくつか買ってきて接続すればシステムができあがる」という訳にはいかない。本論文は、そのような観点に立って、従来のシステムにおいて見過ごさ

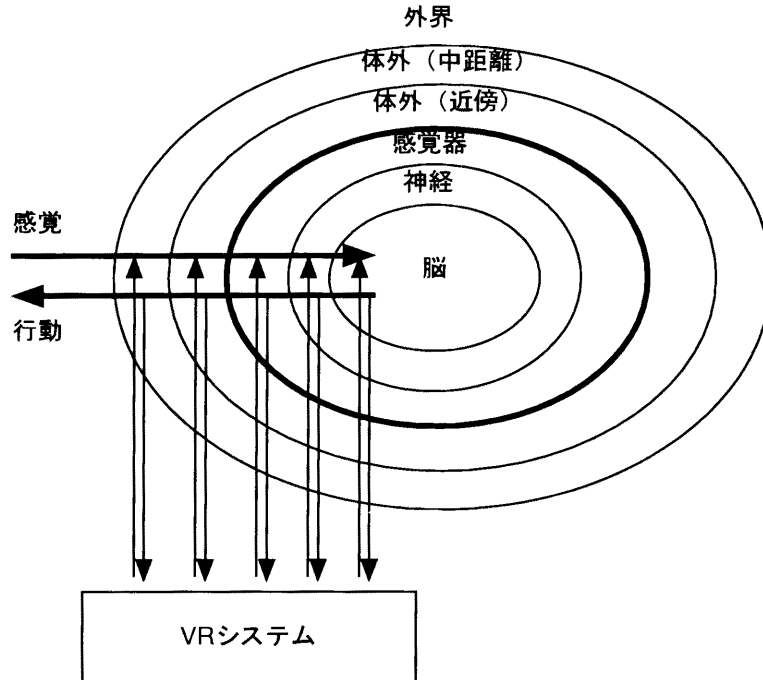


図 1.1 VR システムにおける情報の流れとインタフェースレベル

れてきたものは何かについて考察し，従来のバーチャルリアリティシステムが抱える問題点を，主としてシステムの時間軸方向の品質から解析し，その工学的解決法を探るものである．

1.2 VR およびトレイグジスタンスシステム概観

VR・トレイグジスタンスシステムにおける問題を考察するにあたり，まずシステムの全体像と構成要素について整理する．

1.2.1 情報の流れとインタフェースレベル

人間は，普段外界からの情報を入手し，情報処理を行って行動している．人間の脳の活動を中心において情報の流れを見た場合，外界から脳へ向かうもの（求心性：感覚情報）と脳から外界へ向かうもの（遠心性：制御情報）に分類される．この，感覚情報を得て情報処理を行い，判断して行動することそのものが人間にとっての現実であり，リアリティであると考えられる．バーチャルリアリティとは，この人間の活動の一部を「バーチャルな」もの，つまり実体そのものではなくとも本質的に等価なものと代替することであると言える．これを情報の流れから見ると，遠心性の流れのどこかで制御や行動に関する情報を検出し，外界のシミュレータあるいは遠隔地のロボットなどにより感覚情報を生成し，求心性情報の流れのどこかで割り込んで感覚情報を代替あるいは重畳して人間に提示するものであるととらえることができる（図 1.1）．

外界のシミュレータを用いる場合はバーチャル環境へのトレイグジスタンス（狭義のバーチャルリアリティ），ロボットを利用して遠隔環境とのインタラクションを行う場合は実環境へのトレイグジスタンスに相当する．実環境へのトレイグジスタンスとバーチャル環境へのトレイグジスタンスはユーザにとって本質的に同じことであり [64]，本論文では特に明記しない限り両者を区別せずに扱う．

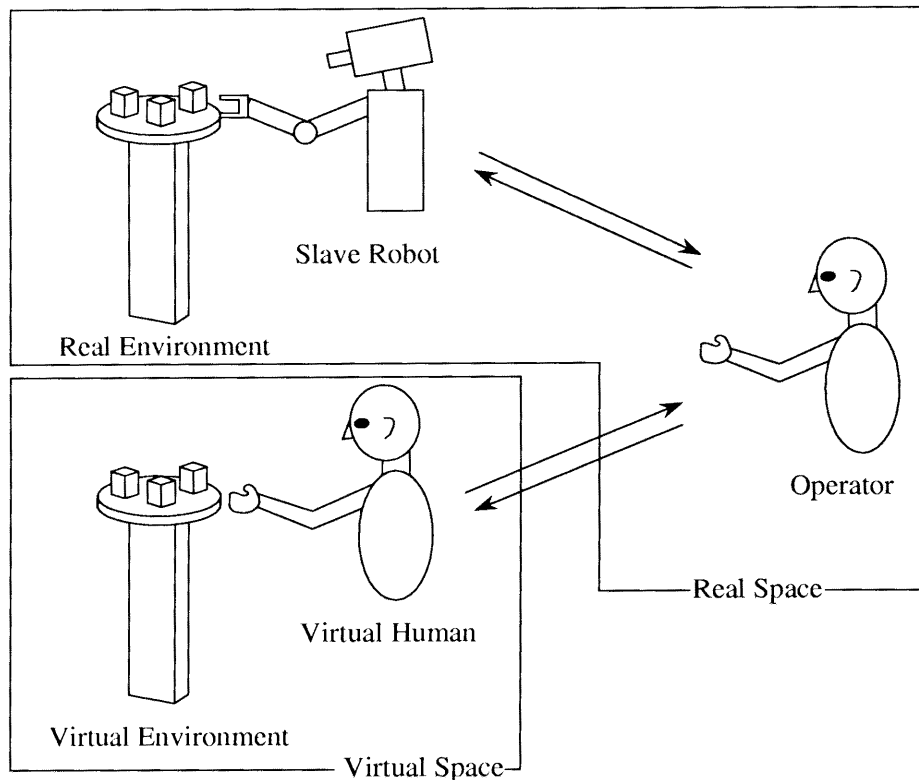


図 1.2 実環境とバーチャル環境へのテレプレゼンス

また、バーチャルリアリティという用語は主にバーチャル環境へのテレプレゼンスのことを指すものとして使用するが、テレプレゼンス全般と同等の意味で使用する場合もある（図 1.2）。

さて、図 1.1 において行動の検出や感覚情報の提示を行う際、どの場所で情報の検出あるいは制御を行うかによって、様々なシステムの形態が考えられる。

- 体内インタフェース：体内の器官と情報のやりとりを行うもの。遠心性情報に関しては、指令が効果器を活動させ行動として発現する以前に情報を取得する。求心性情報については、体内の感覚器と直接あるいはそれすらスキップして感覚情報の流れに割り込む。
 - － 脳インタフェース：脳の活動を計測して人間が何を行おうとしているのかを解析し、また脳の感覚処理細胞へ直接刺激を与えることにより感覚提示を行う。
 - － 神経インタフェース：脳と感覚器や効果器をつなぐ神経を伝達する信号を検出したり、脳へ向かう信号を制御して人工的な感覚を与える。
 - － 体内感覚器/効果器インタフェース：体内に存在する効果器や感覚器と直接やりとりし、効果器が外界に及ぼそうとする影響や人工感覚の提示を行う。
- 体外インタフェース：人間の外側において行動情報の取得や感覚情報のすり替えを行う形態。
 - － 接触インタフェース：装着型の運動計測装置、HMD、力覚や形状を提示する装置などが相当する。
 - － 非接触インタフェース：身体に接触せず情報のやりとりを行う形態。光学式の運動計測装置や包囲型視覚提示装置などが該当する。

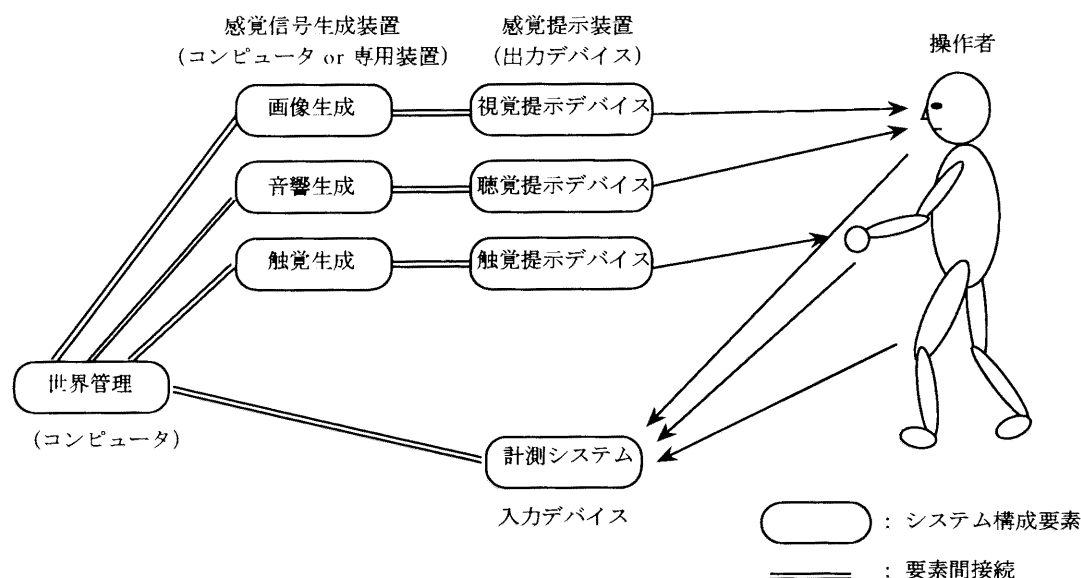


図 1.3 典型的 VR システム

ここで言う体内、体外の区別は厳密に物理的な体内・体外（つまり、埋め込みや侵襲を行うか）というよりも、情報の本質的な受け渡しはどこで行われるかによって分類するものである。

近年、体内における情報の流れを直接検出あるいは制御する技術が開発されつつあり、将来の有望なインタフェースとして期待されている。例えば、脳の活動を計測する EEG, PET, fMRI, MEG などの手法、神経活動のインタフェースであるマイクロニューログラムや神経再生型電極といった技術が挙げられる。しかしながら、これらの技術を人間と機械やコンピュータとのインタフェースとして採用するためには、まだ数々の問題が存在する。すなわち、脳内における活動部位の推定、それぞれの部位における活動と人間の意図する行動との関係の特定、神経活動において検出される信号の同定などの課題が残されている。このため、現時点でのインタフェースは、必然的に人間の体外において、人間の感覚器や効果器を通して行われるものに限定される。

本論文で扱うインタフェースは、現時点で利用可能な体外インタフェースであり、その中でも遠心性情報として動作計測、求心性情報として視覚提示の組み合わせに限定する。この構成によるシステムは従来から数多くの事例が開発されており、バーチャルリアリティ/トレイグジスタンスシステムの典型である。このような典型的なシステムの構成を図 1.3 に示す。システムは「構成要素」（サブシステムあるいはデバイス）と「要素間接続」から構成される。バーチャル世界へのトレイグジスタンスを行う場合、構成要素としては、人間の状態計測を行うセンサなどの入力デバイス、視覚、聴覚、力・触覚提示を行う出力デバイス、それらの感覚信号を生成する感覚信号生成装置、バーチャル世界の状態を管理・シミュレートするコンピュータなどが存在する。実環境へのトレイグジスタンスを行う場合は、遠隔実環境に上記の入出力デバイスを備えたロボットを配置して遠隔環境における操作者のコピーとして振る舞い、バーチャル世界のシミュレーションを行う代わりに遠隔実環境において行動することにより世界への働きかけを行い、環境の情報を収集して操作者に提示する。要素間接続は、これら各種サブシステムにおける入出力およびそれらの間の通信である。

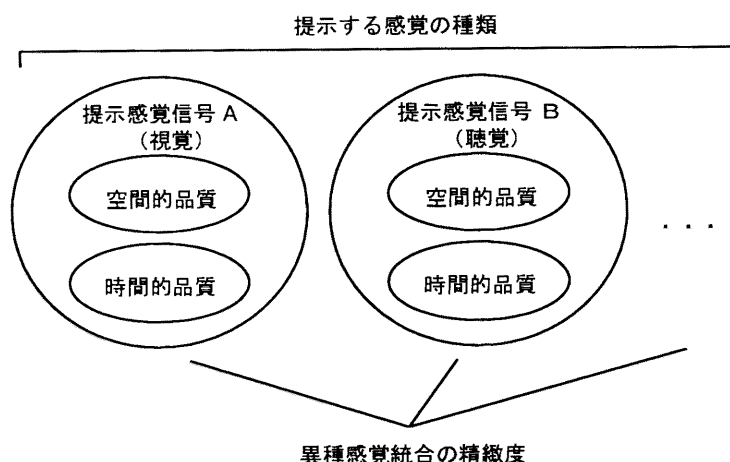


図 1.4 臨場感のための概念的要素

1.2.2 VR システムの品質を決定する要素

バーチャルリアリティおよびテレプレゼンスにおいて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と言われている [64]。

- 1) バーチャル/遠隔環境が自然な 3 次元空間として人間に提示されること。
- 2) 人間がバーチャル/遠隔環境の中で自由に行動でき、人間と環境との相互作用が実時間で行われること。
- 3) 操作する人間自身がバーチャル/遠隔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、操作者の環境への自己投射性の実現されていること。

システムを設計する際には、想定するアプリケーションの規模や提示する感覚信号の品質などを見積もった上で、必要とされるハードウェア/ソフトウェアの性能や機能を決定し、条件を満たす要素を選定（必要ならば設計・製作）することになる。システムとして構成した場合の臨場感に影響を及ぼす概念的要素としては、以下のようなものが考えられる（図 1.4）。

- 提示する感覚の種類：構築するアプリケーションに必要な種類の感覚が提示されること。従来のシステムにおいて最もポピュラーなのは視覚提示であるが、その他にも物体を手で操作する場合には触覚・力感覚提示が有効であり、音による補助や背後の様子などを音で知らせる場合など、聴覚提示が重要な場合も存在する。嗅覚の提示 [19]、味覚の提示は基礎研究が始まった段階であるが、将来的には人間の五感をすべて利用するという意味で重要な地位を占める可能性がある。ここで忘れてはならないのは、明示的な感覚提示を行っていないにもかかわらず、人間が運動を行う場合には常に内部感覚（体性感覚）を伴っていることである。例えば手を動かす場合には目をつぶっていても自分の手先がどこにあるかがわかるし、頭部を動かしたり移動する場合には前庭感覚が関係してくる。
- 空間的品質：計測デバイスの精度、提示する感覚刺激の空間的な「リアルさ」。視覚提示の場合、ハードウェア的にはグラフィックスの解像度、視野角などが該当する。計測デバイスの精度が悪ければ人間の運動が連続に感じられなかったり止まっても「とび」が生じることに

なり、また提示画像の解像度が低ければ近視になったような（あるいは度の合わないメガネをかけているような）感覚が生じ疲労を起こしやすくなる。また、ソフトウェア的には、バーチャル世界をコンピュータグラフィックス (CG) により提示する場合に提示する画像のフォトリアリスティックさなどが該当する。CG を用いて実世界に近い質感を有する画像を生成するレイトレーシングなどの技法は長年研究されており、コンピュータグラフィックス研究における主流である。近年、臨場感のある高品質な画像を提示するために実世界から採取した画像を用いる技術（イメージベーストレンダリング [30] [33] [45] [47] や光線記録法 [37] など）が盛んに研究されており、無機的なコンピュータグラフィックス画像だけでなく実世界に近い画像を提示することが可能になりつつある。

- **時間的品質：** 操作者の運動情報やバーチャル世界の状態の更新レート（毎秒何回か）および時間遅れ（操作者の運動や状態変化が発生してからそれを反映させた感覚提示が行われるまでの全体の経過時間）の双方の性能を確保すること。VR のようなインタラクティブなシステムでは、「ある程度以上」レスポンスが良くなければどんなに精細な感覚信号を提示しても無意味である。時間的品質を上げることはしばしば空間的品質の追求とトレードオフの関係になるが、構築するアプリケーションで最適な妥協点を見つけることは重要である。この意味で、ひたすら画像の高品質を追求するオフラインのコンピュータグラフィックスアニメーションと、VR におけるリアルタイムグラフィックス生成は目的意識が異なります。また、実環境へのトレイグジスタンスにおいては、通信回線を経由することによる遅延が常に問題となるが、物理的な距離に起因する遅延以外の余分な遅延をなるべく生じさせないようなシステムを構築することが重要である。
- **人間の感覚知覚特性への適合度：** 操作者へ感覚情報を提示するにあたっては、人間の感覚知覚特性に適合するように提示を行うことが重要である。例えば、視覚により三次元空間を提示する場合、人間の距離知覚特性は考慮すべき重要な要素であるが、単眼手がかり、両眼手がかりのうちどれを利用するかは押さえておく必要がある。VR 空間中の物体が実際にそこにあるように見えるようにするためには、使用する提示デバイスに適合した画像を提示して、正しい両眼手がかりが与えられるよう注意するべきである。さらに、複数種類の感覚を提示する場合、それらの感覚の間の一貫性（異種感覚統合）が確保されることが重要である。人間は日常的に異種感覚情報を統合して処理しており、この統合メカニズムと矛盾する感覚提示を行うと違和感が生じる。このため、無暗に多種類の感覚情報を提示すればよい訳ではなく、感覚相互の間で矛盾した提示が行われるぐらいならばむしろ提示する感覚の種類を絞った方がよい場合もありうる。明示的には視覚のみの提示を行うシステムにおいても、前述のように頭部運動などによる運動感覚が存在し、視覚との間で統合が行われているため、インタラクティブなシステムでは必ず考慮に入れる必要がある。この要素は空間的・時間的品質と密接に関連しており、例えば頭部搭載型視覚ディスプレイ (HMD) を使用して頭部運動に対応した画像を提示する場合、画像更新レートが小さかったり時間遅れが大きかったりすると、操作者の運動感覚と視覚との間に矛盾が生じ、船酔いのような現象が生じる。

さて、これらすべてが日常生活の時と同じように提示されれば問題ないが、VR システムにおいて感覚提示を行う場合、実際には様々な制約が存在する。例えば、時間的特性に関しては、「人間の運動計測→VR 世界の更新計算もしくは遠隔ロボットの制御→提示用感覚信号の生成あるいはロボット搭載センサによる取得→人間への感覚提示」というループが構成されており、このループの情報伝達が「人間に気づかれない時間の範囲内に」行われることが必要であるが、これらの過程において無視

できない時間遅延は容易に生じうる．そこで、いかにして処理の簡略化を行いつつ、なおかつ「リアリティ」を保つかが問題となってくる．そのためには、システムの中でどの性能を重視しどの性能を低く抑えるか、あるいは別の見方をすると、人間のどの感覚知覚特性に関係する現象を厳密に提示しどの現象の提示を省略するかを適切に決定することがシステム設計にあたっての重要なポイントとなる．つまり、人間の感覚知覚特性の中でも目的に応じて重要な部分とそうでない部分があり、実環境で人間が生活している状況と異なる条件で感覚提示を行っても気づかない、あるいは気づいても容認できる場合が存在する．言うなれば、VR システムではある意味で「だまし絵」的なテクニックを効果的に取り入れることがキーポイントになると考えられる．もちろん、人間の脳はさまざまなレベルでの感覚情報処理を行っており、現象が意識にのぼらない場合でも実は重大な影響を及ぼすことがあるので [54]、気づかないからといって一概に無視してしまうのは危険である．必要・不必要の線引きは VR システムを用いる用途によって異なるため、システム設計者には注意深い考察が求められる．

1.3 運動計測と視覚提示によるシステム

本論文の対象は操作者の運動計測とそれに対応した視覚提示を行うシステムであり、現在最も数多く構築されているシステムの形態である．以下、これらの構成要素について概説する．

1.3.1 運動計測デバイス

- 機械式リンク：リンク機構を用い、先端の位置および姿勢の計測を行う方式．関節角（回転関節の場合）やリンク長（直動関節の場合）を計測し、先端の位置・姿勢を順キネマティクスにより計算する．関節角センサとしてはポテンショメータや光学式エンコーダが用いられる．リンク機構を注意深く設計しないと拘束感を発生させるが、計測原理が単純であるがゆえに高速であるという特徴を持つ．歴史的には最も古くから用いられており、Sutherland が開発したゴニオメータ「ダモクレスの剣」は有名な事例である．また、HMD 開発初期の頃は重量を軽くすることが難しかったため、リンク機構が重力補償として兼用される例が多い．
- 磁気センサ：直交する 3 軸方向に巻いたコイル（ソースコイル）を用いて空間中に磁場を発生させ、やはり 3 軸方向に巻いたセンサコイルにより磁場の強度を計測して、ソースとセンサとの三次元的な関係（位置および姿勢）を算出する．Polhemus 社によりこの方式のセンサが商品化されたことによりバーチャルリアリティが世の中に広まったと言っても過言ではない．その手軽さゆえに非常に数多くのシステムに利用されているが、金属の存在や電磁ノイズにより影響を受けやすく、カタログスペック通りの精度を得るためには部屋ごと設計する必要がある．ノイズや内部的な位置・姿勢算出ルーチンの仕様により、物理的に連続的に移動している状況で計測値が突然大きな飛躍を示す、通称「ポヒマスダンス」と呼ばれる現象を抱える．とはいえ、注意して使用すれば非常に便利なセンサシステムであることには違いない．
- 超音波センサ：発音素子から超音波を発生し、複数の受音素子で音波を計測し、到達時間と受音素子の位置関係などから発音素子の位置を特定する．一般に安価に計測システムを構築できるため、コスト効果を狙った用途に用いられることが多いが、精度がそれほど良くないものが多い．
- 光学式センサ：光学式センサの最大の利点は非接触計測であり、ユーザの拘束感を最小限に抑えられることである．光学式は大別して光点追跡型と画像処理型に分類することができる．光点追跡型で現在までに市販されたセンシングシステムの例としては、スウェーデン Selcom 社に

よる SELSPOT, カナダ Northern Digital 社の OPTOTRAK が挙げられる. 両者とも発光体 (LED) を運動計測対象に装着し, それを複数の視点から追跡 (前者は PSD, 後者はライン CCD を利用) して光点の 3 次元的位置を割り出すものである. 一方, 画像方式は汎用あるいは特殊なカメラにより観察範囲全体を画像としてとらえ, 画像処理により目標物体の抽出と位置の算出を行うものである. この際, 目標物体を容易に検出できるようにするためマーカーを用いることが多い. 一般に, 光点追跡型は高速であるが発光素子に接続されるケーブルを引きずる必要があり, 一方画像処理型は人間にケーブルを這わせる必要はないが計測情報の更新レートがビデオカメラのフレームレート (NTSC 仕様のカメラではフィールド単位の処理で 60Hz) に制限されるという特性を持つ. 光学式全般の欠点は, 発光素子と受光素子との間に遮蔽物が存在すると機能しないことであって, 人間の身体によって遮蔽が生じることは避けられない. この問題の対策としては, 身体側に装着する素子を多数配置し, どのような動きをしても常に複数点の計測ができるようにしておく方法が利用される.

1.3.2 視覚提示デバイス

簡易なシステムでは CRT やプロジェクタに単一の画像を表示することもあるが, 操作者に対して正確な距離感を与えることが必要なトレイグジスタンスシステムや没入型バーチャルリアリティシステムにおいては, 左右両眼にそれぞれ独立な画像を提示するステレオディスプレイが用いられる.

ステレオディスプレイの実現方式としては, 以下のように分類される.

- 空間的分離: ミラーによる折り返しなどを用いて左右用の提示デバイスやスクリーンなどの物理的な配置を分離し, 目の位置から見たこれらの像の全部または一部が重畳して観察されるようにする. あるいは, パララックスバリアやレンチキュラーレンズなどを用いて走査線が左右の目に交互に観察されるような, 空間的インタレースを用いる. 前者は HMD の光学系で多く用いられており, 後者は眼鏡なし立体視システムにおいて多用されている.
- 時間的分離: 画像表示面自体は左右で共有し, 液晶シャッタ眼鏡などを用いて, 左右の目に時間的に交互に画像が観察されるようにする方式. CRT を用いた立体提示や, プロジェクタとスクリーンを用いたシステム, さらにそれを多面化した没入型投影式提示装置 IPT (Immersive Projection Technology) に使用されている.

また, 提示デバイス/スクリーンと操作者の目との位置関係に着目すると, 以下のように分類される.

- 頭部固定型: 提示デバイスを操作者の頭部に対して相対的に固定する方式. HMD はその名の通りこの型であるが, Boom などのように可動覗き眼鏡もこの型に分類される.
- 空間固定型: 画像を提示するスクリーンを頭部に対して固定せず, 周囲の空間に据置にするタイプのディスプレイシステム. 頭部運動に対応した画像を提示するという意味で Head-Tracked Display (HTD) と呼ばれる場合もあるが, HMD を用いたシステムも頭部の計測は行っており, むしろ本質的なのはスクリーンが周囲の空間に固定されており, 操作者の視点とスクリーンとの位置関係が変化するということである. スクリーンの大きさや形態によって, 「覗き窓」のようなもの (CRT を用いた場合) から操作者の周囲をバーチャル/遠隔環境で包囲してしまうシステム (IPT) まで, 様々なバリエーションが存在する.
- オブジェクト指向型 [26]: 頭部固定でもなく, 空間固定でもない, 第 3 のディスプレイ形態. 操作者の周囲の環境を提示するのではなく, 操作者が扱おうとする物体の周囲もしくは全面をスクリーンで囲み, 空間偏在的なバーチャル物体の提示を行うもの.

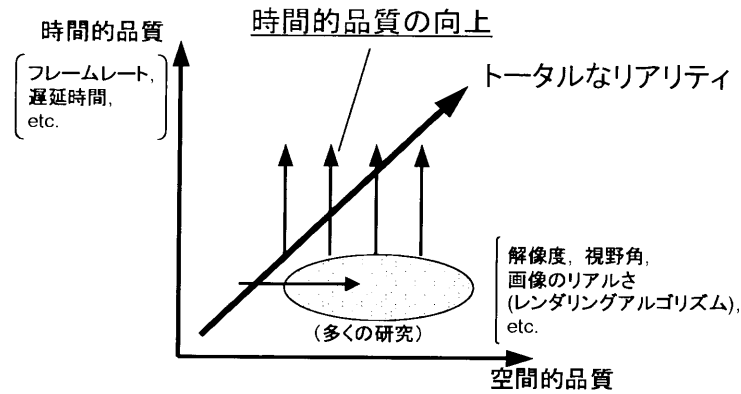


図 1.5 空間的品質と時間的品質

1.3.3 デバイス/サブシステム間接続

システムを構成する要素の間には、情報のやりとりが発生する。これは、テレプレゼンスにおける遠隔サイトとの通信のみならず、ローカルな VR システムにおいても運動計測装置と計算機との間など、サブシステム間の通信が行われている。近年 CPU 性能の飛躍的な発展により従来はリアルタイムでの処理が不可能と考えられていた複雑な処理も次々とリアルタイムに処理可能になりつつあるが、それに比べて入出力チャンネルの速度はそれほど向上していない。このため、こうした通信がシステム全体のボトルネックになる状況は容易に起こりうる。

1.3.4 ソフトウェア

バーチャル環境へのテレプレゼンス、すなわち狭義のバーチャルリアリティにおいては、物理世界をシミュレートする VR 世界管理のソフトウェア、視覚提示用のコンピュータグラフィックス画像生成のためのソフトウェアが必要である。これらは人間の操作・運動に対してリアルタイムに結果を出力する必要があるため、無闇に詳細・複雑な手法を利用することはできない。例えば、物体の変形を扱う場合、一般的なシミュレーションでは有限要素法が用いられるが、これをそのまま用いると計算量が多くリアルタイムでの応答が得られない。このため、何らかの簡略化を施して「ある程度物理法則に忠実な」シミュレーションを行うのが適当である。また、グラフィックス画像の生成に関しては、映画などで用いられる高度に写実的なモデリング・レンダリング技法は適用できず、見栄えをある程度確保しつつ高速に描画する手法が活発に開発・実装されている。

実環境へのテレプレゼンスに関しては、遠隔実世界とのインタフェースであるロボット制御、また実写画像をリアルタイムに転送、表示する画像関係のソフトウェア技術が重要である。

両者に共通するソフトウェア要素としては、操作者側の物理世界と情報世界とのインタフェースである運動計測や、遠隔サイトとの通信、さらにはシステムを構成するサブシステム間の通信などが挙げられる。

1.4 本論文の目的

VR およびテレプレゼンスシステムは、これまで述べたように人間の行動計測、VR 世界の更新計算もしくは遠隔ロボットの制御、提示用感覚信号の生成あるいはロボット搭載センサによる取得、人間への感覚提示、遠隔サイトとの通信といった様々な要素から構成される。その中で、本研究の対

象として扱うのは、行動計測と感覚提示の部分、すなわち人間とシステムとのインタフェースの界面である。また、明示的に提示を行う感覚情報の種類としては視覚を扱うものとする。

視覚提示は VR 研究の歴史において最も古くから多くの研究開発が行われているが、本論文の冒頭で述べたような「違和感」を放置せず、十分なレベルまで引き上げることが肝要であると考える。前述したように、視覚提示のみを行う場合にも、人間の運動を反映するインタラクティブなシステムでは体性感覚（運動感覚）が存在するため、両者の整合性を確保することは自然な VR/遠隔空間提示のために必要不可欠である。そこで、従来の VR システムにおいて運動を行ったときに感じられた「違和感」という漠然とした存在を、視覚と運動感覚との整合性という切り口から見ることにより工学的な解析の俎上に載せ、問題の定式化と解決を図る。

視覚と運動感覚との整合性を確保するためには、人間の運動に対して「即座に」システムが応答しなければならないため、時間軸方向に注意を払う必要がある。一般に、システムのクオリティを高めるには、空間的品質と時間的品質の双方を向上させていく必要がある（図 1.5）。しかるに、これまで多くの研究がデバイスの解像度やフォトリアリスティックなレンダリングアルゴリズムなど、空間的品質の進歩に向けられてきた。これは一つには、空間的品質はデバイス単体のレベルで語ることが出来るのに対して、時間的品質はシステム全体を構築しなければ議論することができないという性質にも起因するであろう。時間的品質に対してあまり注意が払われないと、どれだけコンピュータの速度が進歩してもそのパワーは空間的品質の向上に配分され、時間的品質は不十分なまま一向に改善されないという傾向が続き、そのままでは前述の違和感は解消されない。本論文は、こうした傾向に対して、時間的品質に重点を置くパラダイムシフトを促し、視覚と運動感覚との整合性を確保することにより、「静止している物体や環境がそこに静止して見える」という自然な VR・テレプレゼンスシステムの実現を図る。

1.5 本論文の構成

本章での問題提起に基づき、第 2 章において、運動計測と HMD による視覚提示というごく一般的な構成で、時間的品質に極力重点を置くことにより、運動感覚と提示される視覚との整合性を確保するシステムの構築について記述する。これはいわば現在までの技術でなしうるリファレンスたるべきクオリティを持つシステムの構築を目指したものであり、自然な三次元空間の構成、実時間相互作用、自己投射性という、VR の基本要素を忠実に実現するべく設計・構築したものである。

第 3 章では、HMD を用いたシステムにおいて作業性を損なったり操作者に不快感を与える要素について考察する。従来、HMD を用いた立体視の静的な要素と、純粋な時間遅れに関する研究は行われているが、実際にシステムを組んで頭部運動を行った場合に世界がどう見えるかに関しては言及されていない。このため、HMD 装着時のいわゆる simulator sickness が何により生じるのか不透明なケースが多かった。ここでは、時間遅れの存在しない理想的な場合でさえ、静的な立体視の不整合が存在すると頭部運動を行った場合に動的な世界の変形が知覚されるとともに、不自然な画像の流れにより運動感覚と視覚提示との不整合が生じるという現象を、幾何学的解析により定量的に説明する。同時に、作業実験によって評価を行い、この整合性が確保されることの重要性を示す。このような実験は、第 2 章における低時間遅延システムを構築したことにより初めて可能となったものである。

続いて第 4 章では、近年各地に建設されている没入型投影式提示装置 (IPT) に着目し、頭部搭載型提示装置 (HMD) を用いたシステムとの比較を行うことにより、それぞれのディスプレイにおける利点と欠点を解析する。解析の中心は自然な立体視を実現するための画像生成における透視変換パラメータと、頭部運動、特に回転運動に対して提示されるべき画像の振る舞いであり、前者に関しては IPT が通常のカメラとは異なる特殊な透視変換を必要とすること、後者に関しては HMD が IPT と

比較して厳しい実時間性を要求することを明らかにして、続く2章の出発点とする。

第5章では、HMDを用いたシステムにおける弱点の一つであるシステム全体にわたる時間遅れ、および特にCG画像を用いるバーチャルリアリティシステムにおけるフレームレートが十分に確保できない場合に、操作者に近い場所にローカルな高速ループを構成することにより、等価的に高フレームレートと遅延の補償を行う手法を提案する。

第6章では、IPTディスプレイなど、操作者の頭部運動に対して良好な特性を持つ固定スクリーン型視覚提示装置を、実環境へのトレイグジスタンスに適用するためのシステム構築手法について記述する。

最後に、第7章で、全体のまとめとする。

第 2 章

Virtual TELESAR の構築

本章では，第 1 章で述べた臨場感のためのさまざまな要素を考慮しつつ，バーチャル環境へのテレイグジスタンス（狭義のバーチャルリアリティ）システムの構築に関して記述する．本システムは実環境へのテレイグジスタンスシステム [58] [60] [31] と双対をなすものであり，この概念に基づいて構築された初期のシステム [65] [43] を発展させ，作業環境を含めて実環境版システムと同等の環境を構築したものである．

2.1 システム全体の設計指針

本章で記述するシステムは，実環境版テレイグジスタンスシステムと機能的に等価なシステムをバーチャル環境において実現するものである．実環境版システムでは，頭部運動，上肢運動を含む操作者の運動を計測するマスタシステム（図 2.1），TELESAR (TELEExistence Slave Anthropomorphic Robot) と呼ばれる人間型スレーブロボット（図 2.2），および制御を司る計算機により構成される．このシステムにおいて操作者に提示されるのはスレーブロボット頭部に搭載されたステレオカメラから得られるビデオ画像であり，マスタシステムに装備されている頭部搭載型視覚提示装置 (Head-Mounted Display: HMD) によりスレーブロボットの位置における三次元環境を提示することができる．スレーブロボットは頭部，上肢（マニピュレータ）とも操作者の運動に追従するよう制御され，ロボットの状態が視覚的に操作者へ視覚的にフィードバックされる．ロボットマニピュレータが物体と接触したりハンドが物体を把持することによる力感覚はフィードバックされないユニラテラルなシステムであるが，操作者の運動感覚は常に有効であり，その意味で視覚と運動感覚によるマルチモーダルな性質を持っていると言える．

本章で述べるバーチャル環境版テレイグジスタンスシステムの構築は 1992 年に開始されたが，それまでも HMD を用いた VR システムは数多く存在した．しかしながら，1990 年代初頭におけるそれらのシステムには，しばしば以下のような問題が存在した．

1. 自然な立体視に対する配慮の不足．これは HMD を利用したシステムよりも CRT を利用したシステムに多く見られたが，両眼立体視を行う場合に有効となる立体知覚手がかりを正確に与えるための撮像（画像生成）系・提示系のキャリブレーションが不足で，空間が歪んだり，サイズが不自然に見えるといった例が存在した．
2. VR 空間に反映させる操作者の運動自由度の不足．特に，人間の頭部運動のうち回転 3 自由度のみを利用し，並進運動情報をバーチャル環境に取り入れていないものがあり，体幹運動による頭部位置変化の再現が不可能である．これはすなわち人間の空間知覚手がかりのうち運動視差を無効にする結果となり，操作者の空間把握に悪影響を及ぼす可能性がある．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っており，肘の位置など，人間の腕の

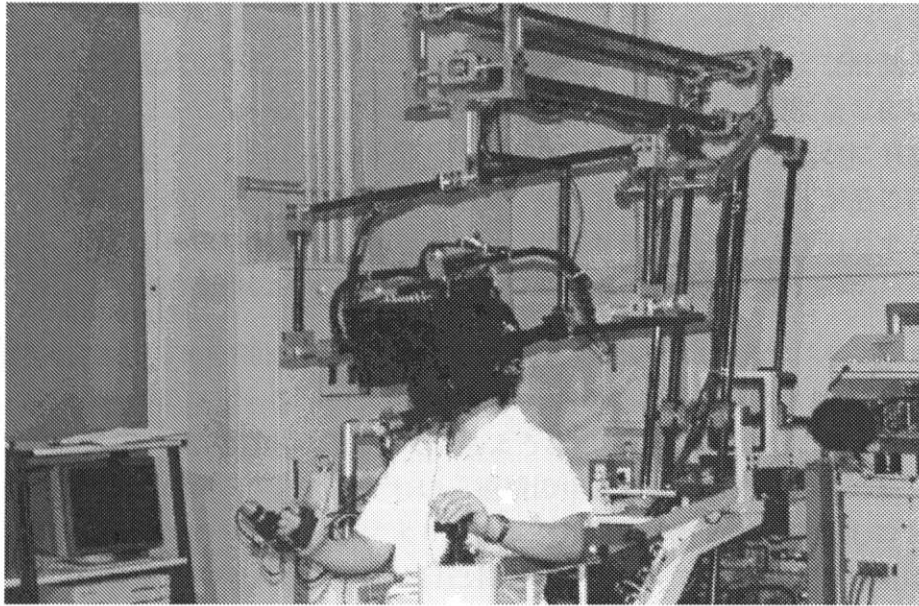


図 2.1 テレイグジスタンスマスタシステム

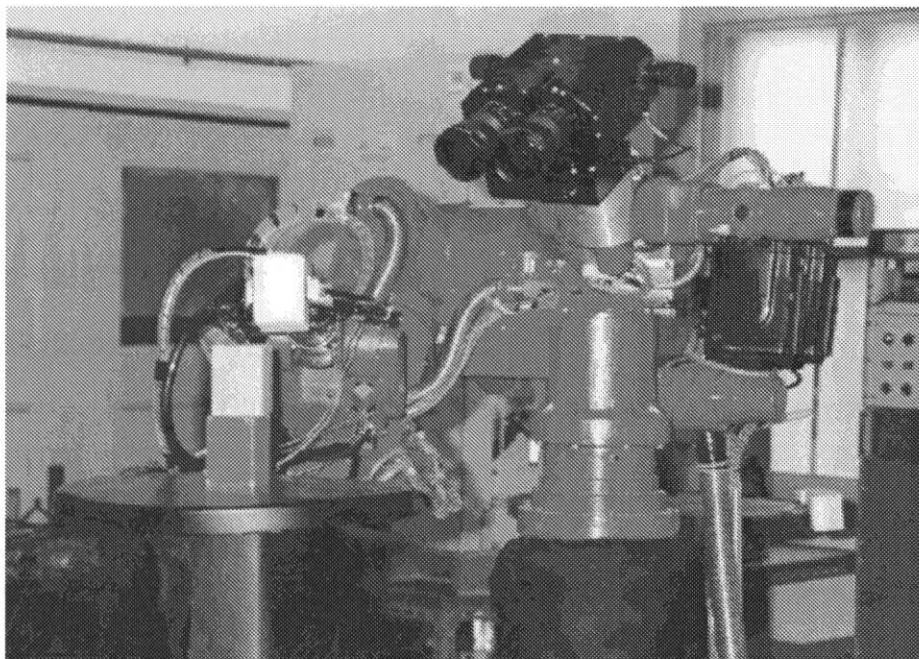


図 2.2 テレイグジスタンススレーブロボット

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バーチャル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、バーチャル環境における自然な動きを実現しているとは言えない．

3. 実時間性に対する配慮の不足．これについてはコンピュータの速度および計測デバイスの性能にも依存するが、コンピュータの CG 画像生成能力に余る複雑・大規模な VR 環境を「無造作に」提示しようとしてフレームレートの低下を招き、頭部を動かすと確かに HMD に表示される画像は変化するものの、その変化が全く連続に感じられないため物体がそこに存在するという知覚が得られない事例が見受けられた．また、センサシステム自体が出力するデータの遅延に無頓着で、ユーザの動作に対する応答が極めて悪いものも存在した．
4. 自己投射性の欠如．VR 空間に「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、操作者自身のコピーをバーチャル環境に存在させるという概念がない．このため、「目」と「手先」がバーチャル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場所のポインティングなどにより「高レベルなコマンド」を発行しコンピュータ側で自動的に環境の操作を実行する場合には自己投射性が乏しくてもさほど不自由はしないが、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．

これらの問題に対し、本章ではハードウェア、ソフトウェアの双方から解決を試みる．まず、ハードウェアに関しては人間の運動を大きな自由度で高速・低遅延・高精度で計測するシステムを使用し [72][73]，さらに計測と画像生成の仕事を分離して互いに待ち時間が発生しないように構成して、システム全体として遅延の最小化を図った [77]．ソフトウェア [74][76] に関しては計測された多自由度情報を VR 空間へ反映させるとともに、画像生成を高速化する仕組みを組み込み、操作者の動作に対する応答性の向上を図った．

2.2 ハードウェア構成

システムのハードウェアを構成する上で最も留意したのは、前述の通り

- 操作者の運動を大きな自由度で計測すること．
- システム全体の時間遅れを最小化すること．

である．本システムは 1992 年に最初の構築を行い [72]，その後コンピュータの高速化やサブシステム間通信の改良 [77] を行なった．

システムの構成の概要を図 2.3 に、システムのブロック図を図 2.4 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、計測用コンピュータに読み込まれる．その情報はバーチャル環境の管理と画像生成を担当するコンピュータへ送られ、バーチャル環境モデルと人間運動の情報をを用いて左眼および右眼用の画像が生成される．この画像を頭部搭載型視覚提示装置 (HMD) に表示し、オペレータにバーチャル環境を三次元的に提示する．

以下、各々の部分について説明を行う．

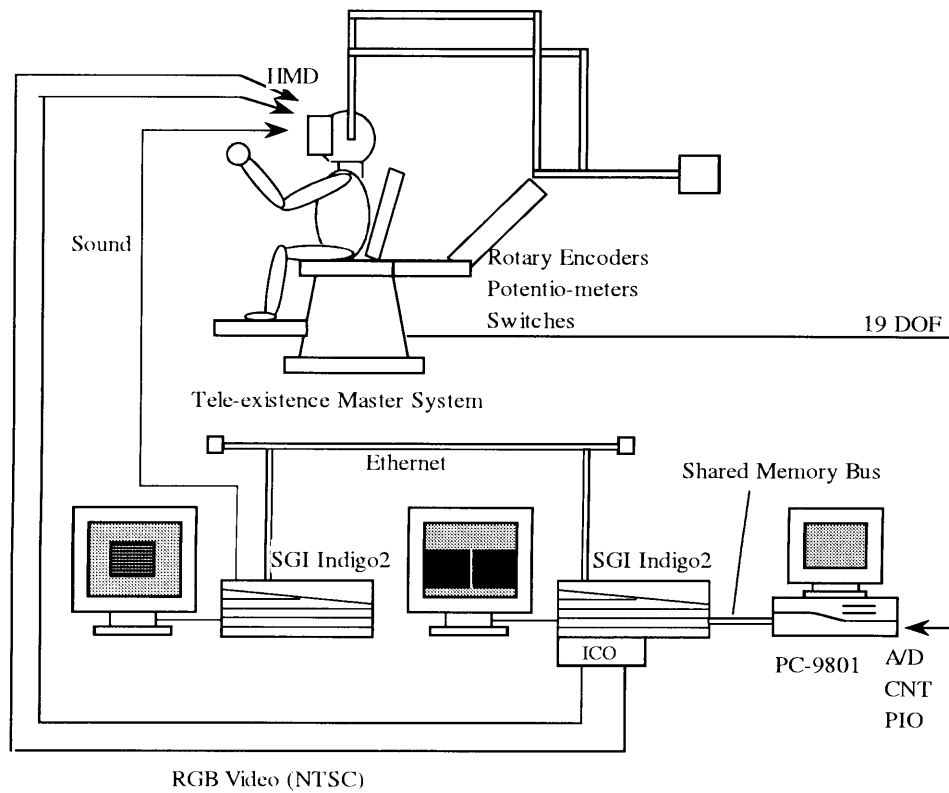


図 2.3 システムの構成

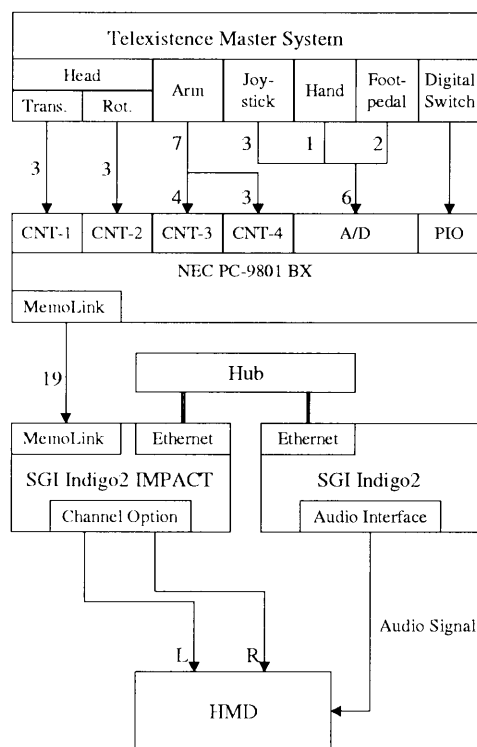


図 2.4 システムのブロック図

2.2.1 人間運動計測部

VR 環境へ操作者の運動を反映させるには、物理世界に存在する人間の運動を実時間で計測することが必要である。VR の分野では、1989 年に磁気を利用することにより点の位置・姿勢を簡単に計測できる装置が Polhemus 社によって市販され、以後 VR における標準的な計測装置として使用されている。この装置は確かに操作者に対する拘束感も少なく有用であるが、その計測原理および実装上の問題に起因する精度や安定性、時間性能に関する限界が存在する。精度・安定性については、ロボットや電子装置が発生するノイズや金属構造物による誤差、非力なプロセッサを用いて角度を算出する際の処理アルゴリズム不備による値の「とび」（通称 Polhemus Dance）などの問題が指摘されている [32]。また、時間性能に関しては、Isotrak など初期の製品で 60Hz、高速版の Fasttrak で 120Hz というデータ更新レートがロボットの制御には不十分であること、計測システム内部において行われるフィルタリングなどの処理による出力データの時間遅れ [35] などの問題が指摘されている。本システムにおいては操作者の運動感覚と視覚との整合性を確保するシステム構築を目的とするため、計測データの安定性と高度な時間性能は必須である。

そこで本システムでは、機械式リンクを利用した計測システムを採用した。このようなシステムとして、実環境へのトレイグジスタンスを実現するために設計・製作されたトレイグジスタンス作業用マスタシステム [31] があり、これを利用することによって 2.1 節で述べた 4 つの問題点のうち、自由度数の問題と実時間性の問題の一部がこの計測システムの持つ特徴によって解決される。

本装置はメカニカルなリンク機構に取り付けられたロータリーエンコーダにより頭部並進運動と回転運動の計測を行っており、高速かつ高精度で計測を行うことが可能である。運動計測の分解能は頭部運動の回転角に関しては 0.025 [deg]、位置に関しては 0.1 [mm] の分解能を確保している。本システムでは、

- a. 頭部運動：並進 3 自由度、回転 3 自由度。平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。

の合計 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）に数値として読み込まれる。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 2[kHz]、時間遅れ 0.5[ms] 以下）。この大きな自由度と高速性により、バーチャル人間の自然な運動と実時間でのバーチャル人間の状態更新が実現される。

計測を行うコンピュータを次に述べる VR 空間管理・画像生成用コンピュータと分離した理由は、一つには遅延時間の最小化のためであり、分離することにより計測の仕事が最高速で行われ、計測のために生じる遅延を極力抑えることを狙ったものである。今一つは、消極的な理由ではあるが、入出力インタフェースボードの適合性である。グラフィックス画像を高速に生成可能なワークステーションはシリアルやごく特殊な通信インタフェースしか接続できず、本システムで使用するロータリーエンコーダ用カウンタや A/D を複数枚装備することが困難である。このため、これらの多チャンネル計測を PC に任せ、ワークステーションとの間で高速通信を行うという構成を採用した。

2.2.2 VR 空間管理・画像生成部

初期の構成では、CG 画像生成の付加を半分に抑えるため、HMD へ出力する左右の画像をそれぞれ 1 台ずつのコンピュータに担当させた。使用したコンピュータは Silicon Graphics（現 SGI）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) であり、これらを Ethernet で接続して通信により同期をとるように構成した。これらのマシンはグラフィックス画像を生成する専用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), パーチャル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。画像を HMD へ出力する際、当初は CRT に表示されたグラフィックス画像をカメラで撮影して HMD に送っていたが、CRT へ表示する代わりに NTSC タイミングの RGB 信号を出力し、直接 HMD へ送るように変更した。

画像生成用コンピュータは、後に CPU およびグラフィックスハードウェアが高速化された Indigo2 High IMPACT (CPU R4400 200MHz) に置き換えた。グラフィックスエンジンの性能は毎秒 38 万四角形ポリゴンである。この構成では左右両眼用画像を 1 台で生成し、同時に出力する。このワークステーションには複数の画像信号を出力可能なボード (IMPACT Channel Option: ICO) を付加しており、両眼提示用画像を 1 台で生成・出力できるようになっている。ICO ボードは NTSC タイミングの画像信号を最大 4 画面分出力できるが、本システムではそのうち 2 画面分の出力を利用している。

2.2.3 信号と情報の流れ

計測用計算機 (PC) からパーチャル環境構成・画像生成用計算機へは、当初シリアル回線 (19200 bps) を用いて運動情報の転送を行った。トレイグジスタンスマスタシステムにより計測される 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 回程度であり、画像生成用コンピュータを高速化するとデータの伝送がシステム全体のボトルネックとなることが懸念された。このため、計測用コンピュータと画像生成用コンピュータのそれぞれに共有メモリボード (インターフェース社製 MemoLink) を装着して互いに接続した。この結果、データの計測データの更新レートは 2[kHz] まで向上した。この更新レートは計測用コンピュータ上でカウンタボードからの読み込み、カウンタパルス値から物理的角度値への変換、および変換結果のメモリボードへの書き込みに要する時間であり、遅延は 1 サイクル周期 (0.5[ms]) 以内である。

この計測データ更新レートは、パーチャル環境画像を数十 Hz で生成する目的だけのためにはオーバースペックであるが、本システムでは操作者の運動感覚と提示される視覚との整合性を確保するため、システム全体の時間遅れを極力小さくすることを狙ってこのような構成とした。その結果、画像生成プログラムがデータを必要とした時に常に最新の計測データが同じ計算機のメモリ上に用意されているようになり、画像生成・表示に起因する遅延時間 (フレームレート毎秒 60 枚の場合、最小で 2 フレーム分の 1/30 [s]) と比較して、データ計測・転送に起因する遅延時間は極めて小さく、無視できる範囲に収まっている。

2.2.4 画像の提示

マスタ装置のヘルメットに固定された HMD[31] を使用した。視野角は水平 40 [deg.] で、解像度 720 × 240 のカラー TFT 液晶パネルを使用している。LCD パネルは視点の前方 1 [m] の位置に結像されるよう光学系が設計されている。

3. 高速であること．オペレータの運動がバーチャル人間の運動にリアルタイムに反映され、オペレータとバーチャル環境との相互作用が実時間で生じるために必要である．時間的な品質には、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、船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、バーチャル世界の状態更新、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．
4. 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、オペレータがバーチャル環境の中へ入り込んだような感覚を与える．
5. 自然な3次元バーチャル空間が構成されること．バーチャル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．同時に、バーチャル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が必要である．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、バーチャル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、バーチャル人間のスケールリングを注意深く行なう必要がある．
6. バーチャル人間の感覚器が捉えた感覚情報が、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．バーチャル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．

以上の要請のうち、最初の3点についてはVR空間を構成するソフトウェアの仕組みとしてシステムティックに導入することが求められる．その他は、ソフトウェアの機構というよりは、VR空間のモデリングとアプリケーションにおけるパラメータ設定の問題である．

1990年代後半になっていわゆるVR用ソフトウェアパッケージが複数市販されるようになったが、これらのソフトウェアが行う仕事には

- VR空間の管理と画像生成を行う基本的な枠組みを提供し、VR空間構築の作業においてそれほどプログラミングを意識せずに行えるようサポートする
- VRシステムにおいて使用される各種入出力デバイスの使用法をパッケージ化し、デバイスとのやりとりに使うコマンドやコンピュータのI/Oプログラミングをカプセル化することによりこれらのデバイスを簡単に使用できるようサポートする

という二通りの側面がある．

1980年代まではコンピュータグラフィックス画像を生成するためのAPI（アプリケーションインタフェース）すらシステムごとにまちまちであり互換性がなかったが、幸いにして本システムを構築

する時点ではある程度標準化が進み、プログラミング環境が提供されていたため、本システムでもそれを利用した。本システムで使用する Silicon Graphics 社のワークステーションは VR の分野でほぼ標準と言ってよいプラットフォームであり、同社のワークステーション上では IRIS GL [55] という三次元グラフィックスの API が提供されている。これは後により多くのプラットフォーム上で動作し完全な業界標準となる OpenGL [38] の母体となった API であり、OpenGL には IRIS GL の基本的な構造が受け継がれている。

さて、本システムにおけるソフトウェアの仕事は、オペレーティングシステム (OS) およびグラフィックス API を利用して、コンピュータ外部から操作者の運動情報を取り込んで VR 世界のモデルを管理・更新し、画像を生成して出力することである。これを行うために、基本的な仕組みを提供する部分はライブラリの形で体系化し、アプリケーション依存の部分は直接プログラムで記述するという方法を採用した。ただし、分散処理や外部との入出力は基本的な仕組みの中でも外部との入出力を行う部分はハードウェアシステムの構成に強く依存するものであり、前述の「デバイスサポートライブラリ」を作成することは本研究にとって本質ではないため、これもアプリケーションプログラムに直接記述した。体系的にライブラリ化を行ったのは VR 空間の管理と画像生成に関する部分である。グラフィックス API は座標変換やポリゴン描画といったレベルの仕事を担当するものであり、そのままでは VR 空間の管理に向いているとは言い難い。

後述するように、VR 空間の系統的な管理を実現するにはソフトウェア工学で言うところのオブジェクト指向を導入するとよい。開発を始めた当時、Silicon Graphics 社から IRIS Inventor というツールキットがリリースされたが、初期のバージョンでは実行速度が遅く、単に CAD モデルをコンピュータのモニタ上に表示するには利用できるが、実時間性が最優先される VR のソフトウェアとしては満足できるものではなかった。このため、本システムでは独自にこのレイヤーのソフトウェアを製作することとした。ちなみに、IRIS Inventor の後のバージョン、すなわち Open Inventor では、本章で記述するオブジェクト・カリング、Level Of Detail などの高速化のための仕組みが組み込まれた。これらの概念は実時間指向のシミュレーションツールキットである IRIS Performer には当初から組み込まれたが、本システムの開発当時はまだ利用できる段階になく、また Performer はデスクサイド以上の大型（かつ高価）なシステムをターゲットとしていたので、研究室レベルで利用可能なワークステーションで動作可能なツールの構築は必須であった。

2.4 VR 空間管理・画像生成用ライブラリ

本節では、VR 空間表現の基礎事項について概説した後、本システムで構築した VR 空間構築用ライブラリにおける技術的なポイントと実装について述べる。

2.4.1 VR 空間における物体の記述方法とシーングラフ

VR 空間における物体の記述と配置

VR 空間中の物体は、通常多面体としてモデル化される。より滑らかな曲面を表現するためには NURBS などの表現が用いられるが、現在までのグラフィックスシステムにおいてはこうした曲面表現も最終的により多数の点の集合として分解されるので、頂点による表現が本質的である。多面体の頂点は、ある座標系を基準とした 3 次元ベクトルとして表現される。基準座標系を $\{A\}$ とすると、

$${}^A P = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \quad (2.1)$$

と記述される．このベクトルを縦ベクトルで記述する流儀と横ベクトルで記述する流儀が存在し，コンピュータグラフィックス関係者は横ベクトルで記述し，ロボット工学関係者や一般的な数学書では縦ベクトルで記述する傾向がある．IRIS GL では横ベクトル方式で記述を行っていたが，Open GL でも縦ベクトル表記に統一され，本論文でも縦ベクトル表記を使用する．

物体を空間中に配置するには，空間全体の基準となる座標系を定義し，物体座標系との位置関係を記述する [4]．一般に，座標系同士の関係は回転と平行移動により定義される．ある座標系 $\{A\}$ を回転して得られる座標系 $\{B\}$ は，次の回転行列を用いて表現される．

$${}^A_B R = \begin{bmatrix} {}^A\hat{X}_B & {}^A\hat{Y}_B & {}^A\hat{Z}_B \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (2.2)$$

ここで， ${}^A\hat{X}_B, {}^A\hat{Y}_B, {}^A\hat{Z}_B$ はそれぞれ座標系 $\{A\}$ を基準とした座標系 $\{B\}$ の座標軸のベクトルである．また， $\{A\}$ を平行移動して得られる座標系 $\{B\}$ は，座標系 $\{B\}$ の原点を座標系 $\{A\}$ に関して記述することで特定でき，これを ${}^A P_{Borg}$ と表記する．

回転と平行移動を統合して表記しその組合せを単一の演算子として扱うには，頂点ベクトルに冗長な項を付加して 4×1 のベクトル

$${}^A P = \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} \quad (2.3)$$

の形で表記しておき， 4×4 の同次変換行列を用いる．

$${}^A_B T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & x \\ r_{21} & r_{22} & r_{23} & y \\ r_{31} & r_{32} & r_{33} & z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^A_B R & {}^A P_{Borg} \\ \mathbf{0} & 1 \end{bmatrix} \quad (2.4)$$

上式において， \mathbf{R} は姿勢を表す 3×3 行列， \mathbf{p} は位置を表す 3×1 ベクトルである．

さてここで，図 2.6 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 に付随する座標系

$\{S\}$: A の部分物体 S に付随する座標系

$\{A\}$ 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

$${}^O P_A = {}^O_A T {}^A P_A \quad (2.5)$$

である．

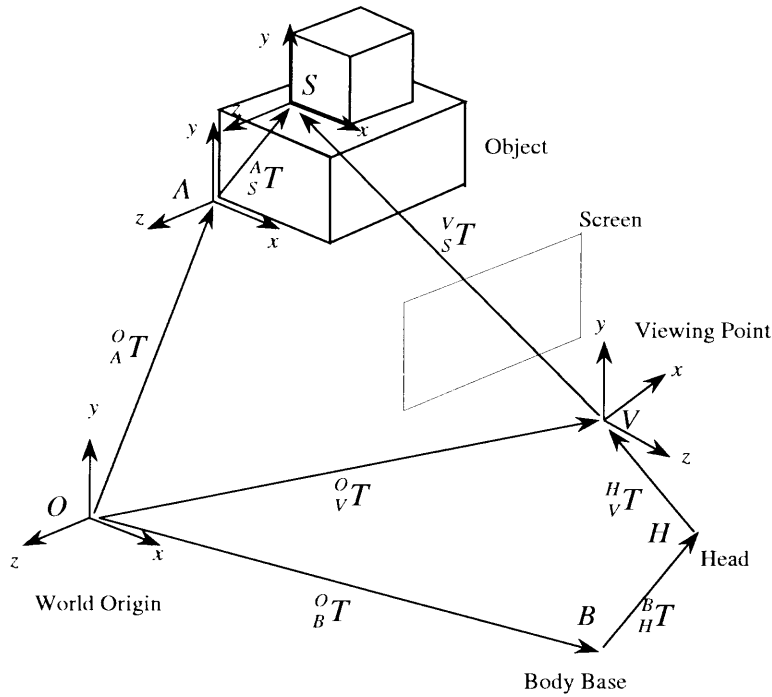


図 2.6 座標系の記述と階層的 world 構築

次に，物体 A がいくつかの部分要素から構成される場合を考える．図 2.6 において，ある部分構成要素 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

$${}^A P_S = {}^A_S T {}^S P_S \quad (2.6)$$

であるから，世界基準座標系に関しては

$${}^O P_S = {}^O_A T {}^A P_S = {}^O_A T {}^A_S T {}^S P_S = {}^O_S T {}^S P_S \quad (2.7)$$

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．上式において， ${}^O_S T (= {}^O_A T {}^A_S T)$ が物体 S の世界座標系に対する位置・姿勢を表す変換行列となる．

階層的 world 構築とシーングラフ

このように，VR 空間中に存在する物体を階層的に管理すると，複数の物体の集合体をあたかも 1 つの物体であるかのようにまとめて扱うことができ便利である．こうした階層構造を構築する場合，ツリー構造が用いられる．グラフィックス命令をツリー構造のノード上に配置したものを，シーングラフと呼ぶ．本システムの構築時点では，こうした階層構造を用いて VR 空間管理を行うソフトウェアツールはまだ一般化しておらず，複数の物体をただ並列に列挙するものが多かった．それには，グラフィックスハードウェアの能力が貧弱で複雑な VR 空間を実時間表示できなかったという背景も考えられる．その後グラフィックスハードウェアの高速化および低価格化によって 3 次元 CG 画像の実時間表示が普及するにつれて VR 空間の構造化に対する需要が増大したこともあり，現在までに複数のプログラミングツールキット，モデリングソフトウェア，VR 用パッケージソフトウェアにおいてシーングラフの概念が利用されるようになった．例えば，SGI 社のワークステーションにおけるツー

ルキット IRIS Inventor およびその機種依存性を排除して複数のプラットフォーム上で動作可能にした Open Inventor, 同社の実時間シミュレーション用ツールキット IRIS Performer, Open Inventor をベースにした VRML (Virtual Reality Modeling Language) などである。現在ではこれらのツールキットやパッケージソフトウェアを利用して VR アプリケーションを構築可能であるが、本システムのライブラリはこれらとは独立に VR アプリケーションに求められるソフトウェアの枠組みを考察し、設計・実装したものである。

上記の一般的なコンピュータグラフィックス分野におけるシーングラフはあくまでグラフィックス命令の並びを構造化・整理することを目的としており、同じ階層に属するノードの間に順序依存性が存在することがある。例えば、物体表面材質の指定とそれを適用して描画される物体ノードが並列に配置される。これは、「材質を一回指定したらそれを変更しない限り同じ材質を用いて描画を行う」という 3 次元グラフィックス API における構造をそのまま引きずっているものであり、描画実行の効率化の観点からは容認されうるが、VR 空間の構造化という意味では非常に奇妙な構造である。なぜならば、物体の材質は物体に付随する属性であり、材質だけが VR 空間中に存在するということはないからである。また、本研究の範囲を超えるが、VR 空間における力・触覚提示を行う場合には物体に付随する属性として質量、表面インピーダンスなどを付加することが考えられるが、上記のようにグラフィックス API の構造に依存したシーングラフではこうした力・触覚用の属性の指定と整合性を保つことが非常に困難になる。

これに対し、本システムにおけるシーングラフは、最初から VR 空間の体系的な記述を目的とするものであり、グラフィックス命令ではなく VR 物体そのものをノードとして配置するというアプローチをとった。このため、本方式の適用可能性は視覚提示だけにとどまらず、実際、本システムで開発した VR 空間管理用ライブラリは、力・触覚による形状提示と視覚提示を同時に行うシステム [14] [22] における統一的モデル記述のベースとして採用されている。

視点の指定と画像生成

バーチャル環境の物体上の点は、モデリング変換（バーチャル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、視点をバーチャル人間の左・右眼の位置におくことが重要である。

ビューイング変換には、オペレータの体のバーチャル空間内での位置・姿勢、体と頭部の位置・姿勢の関係、頭の中心と左眼/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座標系への変換行列は

$${}^O_V T = {}^O_B T {}^B_H T {}^H_V T \quad (2.8)$$

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。 ${}^O_S T$ を視点座標系を経由して表現すると

$${}^O_S T = {}^O_V T {}^V_S T \quad (2.9)$$

と表されるので、視点座標系を基準とした物体 S の変換行列は

$${}^V_S T = ({}^O_V T)^{-1} {}^O_S T \quad (2.10)$$

となり、物体座標系 $\{S\}$ 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。

2.4.2 オブジェクト指向の導入

バーチャル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構造化せずに行なうのは非常に複雑でかつ見通しの悪い作業になる。プログラミング言語の観点からは、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、バーチャル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（有効範囲）内で無制限に行なうことができる。これは一見便利ようであるが、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プログラマの不注意によりこれが崩される可能性がある。また、手続き型言語においてはこれらのデータのスコープの選択肢は関数内（ローカル変数）、ソースファイル内（static 変数）、および大域的（グローバル変数）しかなく、別のファイルからデータにアクセスするためにはグローバル変数として定義せざるを得ないが、規模が大きくなると変数名が互いにぶつからないよう名前空間を系統的に管理することが著しく困難になる。

このような理由から、バーチャル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フェースが決められ整理されていることが望ましい。そのためには、ソフトウェア工学で言うところの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。物体に付随するデータへアクセスするには、その物体を表すクラスのメソッド（インタフェース）を通して行うことにより、データ管理における整合性と見通しの良い名前空間が実現される。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 [34] 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、実時間での人間オペレータとバーチャル環境の相互作用を意図しているものではない。本章で記述するバーチャル環境構成法は、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。

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、効率、機動性、一般性などの観点から、C++ 言語のクラスを適用した。本システムには、バーチャル環境構成法は、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することにより単体としてのバーチャル物体を記述し、それらをグルーピングしたオブジェクトを定義して個々の物体を配置していくことにより空間全体を記述するという方法をとった。本方式は、形状データを独立に用意しておいて関数呼び出しによりバーチャル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フェースを統一的に記述していく点に特色を有する。

シーングラフの節でも述べたように、本システムの設計思想として、グラフィックス描画プログラミング手法としてのオブジェクト指向だけではなく、文字通り VR 物体をオブジェクトに対応させている。このため、オブジェクト指向言語が持つ特徴をそのまま VR 物体に対して適用することが可能になり、以下のような利点を実現している。

- データとインタフェースの系統的管理：データのカプセル化により、VR 空間中に存在する物体の配置、物体が持つ属性に関するデータが系統的に管理され、物体操作のためのインタフェースが整理される。物体に付随する属性などのデータは、その物体を表すクラスが用意するインタフェースを通して行い、予期しないデータの整合性喪失や名前空間におけるバッティングを避けることができる。
- VR 物体の部品化：単なるグラフィックス命令の羅列ではなく、VR 物体を「オブジェクト」として定義できる。VR 物体をクラスとして定義することにより、物体の複製（複数のインスタ

ンスの生成)や物体の性質の継承などが簡単に行える。

- プログラミングの省力化: パーチャル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パーチャル環境を構築することができる。また、VR空間の管理をサポートすることを目的として、演算子を多重定義することにより3次元座標や座標変換行列などを普通の整数や実数などと同じように直観的な演算子を用いて計算を行なうことができる。
- 拡張性: オブジェクト指向で言うところの継承により、基本クラスから派生させた新しいクラスを定義して、VR物体に関する属性や振る舞いの追加などを容易に行うことができる。これは、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において有効である。

2.4.3 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描画処理の効率化は必須であり、本ソフトウェアシステム構築の大きな目的の一つである。本システムにおいて導入した手法は、幾何コンパイル、オブジェクト・カリング、Level Of Detail であり、以下これらの手法について記述する。

幾何コンパイル

2.4.1 節で述べたように、それぞれの物体で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ておき、シーングラフを構成するツリーの根元から順番に変換行列をかければ物体のグローバルな位置および姿勢が計算できる。しかし、動かない物体、あるいは物体グループの中で位置関係が変化しない要素の場合、毎回これを計算することは無駄である。そこで、このような物体についてはあらかじめ定めた基準座標系(場合によっては世界座標系)からの位置・姿勢を計算しておく機能を導入する。本システムにおいては、これを幾何コンパイルと呼ぶ。表示物体の数が多い場合には、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。

こうした幾何学的な位置関係(ジオメトリ)計算量を低減に貢献する類似の手法としては、OpenGL API に組み込まれているディスプレイリストの考え方がある。これは、回転、平行移動などの座標変換、色などの物体属性の指定、ポリゴン描画の実行など多岐にわたるグラフィックス命令をあらかじめグラフィックスシステムに登録しておくものである。これを利用すると、特に回転行列の指定において登録時に三角関数を実行して結果の同次変換行列の形で記憶しておくので、実行時には単に行列の乗算だけで済み、ジオメトリ計算の高速化が図られる。しかし、この方法では多岐にわたるグラフィックス命令をリストとして登録するため、一連の座標変換と他のグラフィックス命令が混在する形となり、座標変換だけの変更により登録したリスト全体が無効になるという無駄が生じる。また、ディスプレイリストでは個々の座標変換命令に対する同次変換行列の展開は行われるが、階層構造の何段にもわたって行列をまとめる作業は行われない。その点、幾何コンパイルの手法はジオメトリ計算だけを扱うので、他のグラフィックス命令と干渉しない利点を有し、また、深い階層構造においては基準となる座標系から末端の物体に付随する座標系までのジオメトリ計算が一回の行列乗算だけになるため、より効率的である。

幾何コンパイルの手順を以下に示す。ここでは、幾何コンパイルの原点(通常は世界座標系)を $\{A_0\}$ 、以下世界のツリー構造を下にたどるにつれて、それぞれの物体に付随する座標系を $\{A_1\}$, $\{A_2\}$, ..., $\{A_{n-1}\}$, $\{A_n\}$ とし、対応する(ローカルな)変換行列をそれぞれ ${}^{A_0}_{A_1}T$, ${}^{A_1}_{A_2}T$, ..., ${}^{A_{n-1}}_{A_n}T$ とする。

1. 幾何コンパイル用変換行列 M_C に、ローカルな変換行列を代入する.

$$\begin{aligned} i &= n, \\ M_n &= \begin{matrix} A_i \\ A_i \end{matrix}^{-1} T \end{aligned} \quad (2.11)$$

2. 親物体が存在するかどうか調べ、存在しなければ終了（世界座標系に到達）．または、親物体との間に幾何コンパイルを行うかどうかの設定を調べ、行わない設定となっていれば終了．
3. 親物体が幾何コンパイルされているかどうか調べる．幾何コンパイル済みであれば、親物体のコンパイル済みの行列をかけ、終了．

$$M_n = M_{i-1} M_i \quad (2.12)$$

4. 親物体の幾何コンパイルが行われていなければ、親物体のローカルな変換行列をかける．

$$M_n = \begin{matrix} A_{i-2} \\ A_{i-1} \end{matrix} T M_n \quad (2.13)$$

5. $i := i - 1$ として、2. に戻る．

この操作が終了すると、物体の幾何コンパイル用変換行列には

$$M_n = \begin{matrix} A_0 \\ A_1 \end{matrix} T \begin{matrix} A_1 \\ A_2 \end{matrix} T \cdots \begin{matrix} A_{n-1} \\ A_n \end{matrix} T \quad (2.14)$$

という値が保持されている．

世界全体に対して固定された物体の場合は幾何コンパイル基準座標系を世界座標系にとっておけばよいが、運動する物体の場合は、最初に一度幾何コンパイルを行なっても、その位置や姿勢が変化してしまえば前に行なった幾何コンパイルの結果は無効になる．このとき、位置や姿勢が変化する度に世界座標系に対するその物体の幾何コンパイルを行なっていたのでは、毎回その物体のツリー構造をたどって末端のオブジェクトの一つ一つまで変換行列の再計算を行なう必要があり、特にその物体が複雑な構成要素から成り立っている場合にはかえって非効率であるといえる．

そこで、このような世界の中で運動する物体の描画を効率良く行なうため、幾何コンパイルの基準となるフレームを世界原点以外の任意の物体に付随する座標系にとり、幾何コンパイルの計算を局所的なツリーの中にとどめておく機能を用意する．これは、各々の物体が親物体に対する幾何コンパイル計算を行なうかどうかを示すフラグを持つことによって実現できる．上記の手順2において、親物体が存在するにも関わらず幾何コンパイルを行わないということは、親物体との間の位置関係が頻繁に変更される場合であり、世界全体に対してではなく物体内の部分的な位置関係の計算にとどめることにより、幾何コンパイルの頻繁な再計算を避けることができる．例えば、ロボットアームの場合、各リンクを構成する部品同士的位置関係は変化しないが、リンク同士は関節変数の変化に応じて頻繁に更新される．このとき、リンクを構成する部品の幾何コンパイル基準座標系をそのリンクを代表する座標系に定めておけば、それぞれの部品に対する幾何コンパイル再計算を行わずに済む．

幾何コンパイルの基準座標系から該当物体までのツリー構造の中でどれか1箇所が移動、回転などの変更を受けた時は、変更された物体の下の物体の幾何コンパイルはすべて無効になる、それ以降明示的に幾何コンパイルが行なわれるまでその物体に付随する座標系の基準座標系に対する座標変換行列の計算は行なわれない．

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基準座標系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．

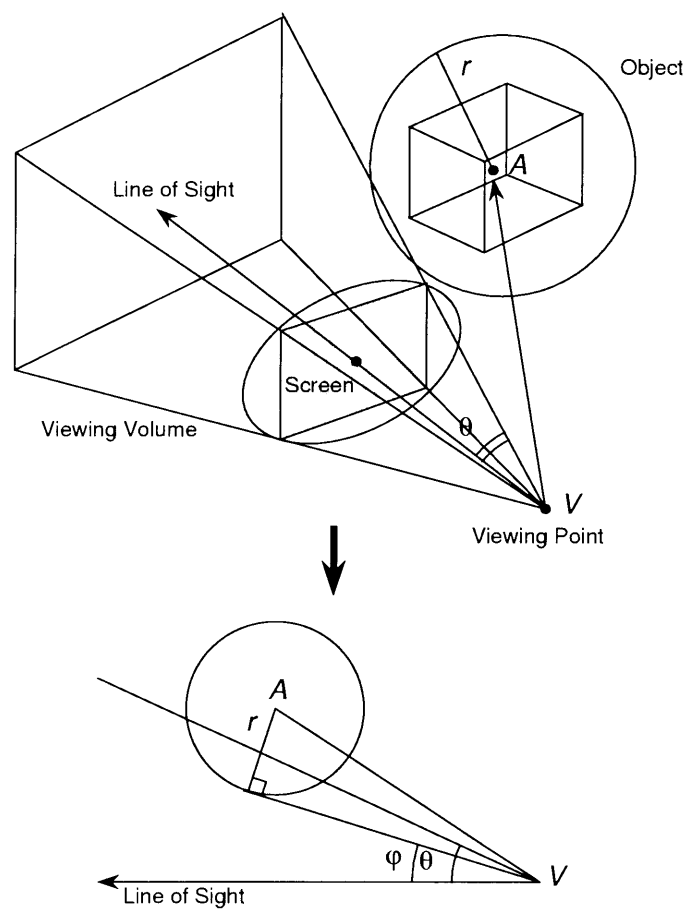


図 2.7 視円錐と包絡球によるオブジェクト・カリング判定

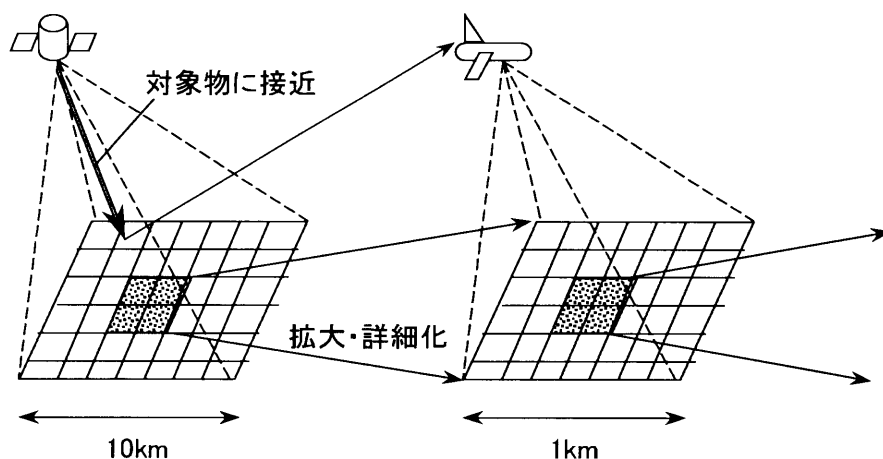


図 2.8 Level Of Detail 制御の概念

オブジェクト・カリング

CAD データの表示などにおいてはバーチャル物体のモデル全体を色々な方向から眺めるため、モデル化されている物体全部を表示する。しかし、VR 空間内を動き回るウォークスルーなどのアプリケーションにおいては、視野に入ってくる物体は全体の中のごくわずかである場合が多い。視野内に入らない物体の描画をスキップする処理は通常グラフィックス生成におけるレンダリングステージの直前にビューイングボリューム（視体積）によって行われるが、その段階以前にも頂点計算などが行われており、視野外の物体に対して描画命令を発効することは無駄が多い。そこで、視点の位置および方向、物体の位置およびお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しておけば、描画処理の高速化につながる。この手法は、オブジェクト・カリング (object culling) と呼ばれる。

この手法において、物体の描画実行・除外の判定処理が複雑で計算負荷が高いようでは、本末転倒である。本システムでは、最も簡単な視円錐と包絡球によるオブジェクト・カリングの機能を提供している。図 2.7 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わない。

オブジェクト・カリングの効果はアプリケーションの形態と構築する VR 空間に強く依存するが、ウォークスルーアプリケーションなどにおいて最も有効に機能する場合、グラフィックスシステムへ送り込む命令の量を桁違いに低減することができ、高速化への寄与は絶大である。

Level Of Detail

大規模な空間の表示を行う場合、表示対象モデルのポリゴン数および物体表面にテクスチャ画像として貼り付ける画像データは膨大な量になる。このようなモデルの詳細部分まで表示しようとすると描画に時間がかかり、フレームレートが低下して操作者の動きに対する応答が悪化する。

この問題に対処するための一手段として、表示する物体モデルの詳細度をコントロールする Level Of Detail という概念が適用できる。詳細度の異なるモデルあるいは詳細度可変のモデルを用意しておき、操作者の視点から物体までの距離や角度などの見えかたのパラメータに応じてどの程度詳細な表示が求められるかを評価して、必要最低限の詳細度のモデルを用いて画像生成を行うものである。

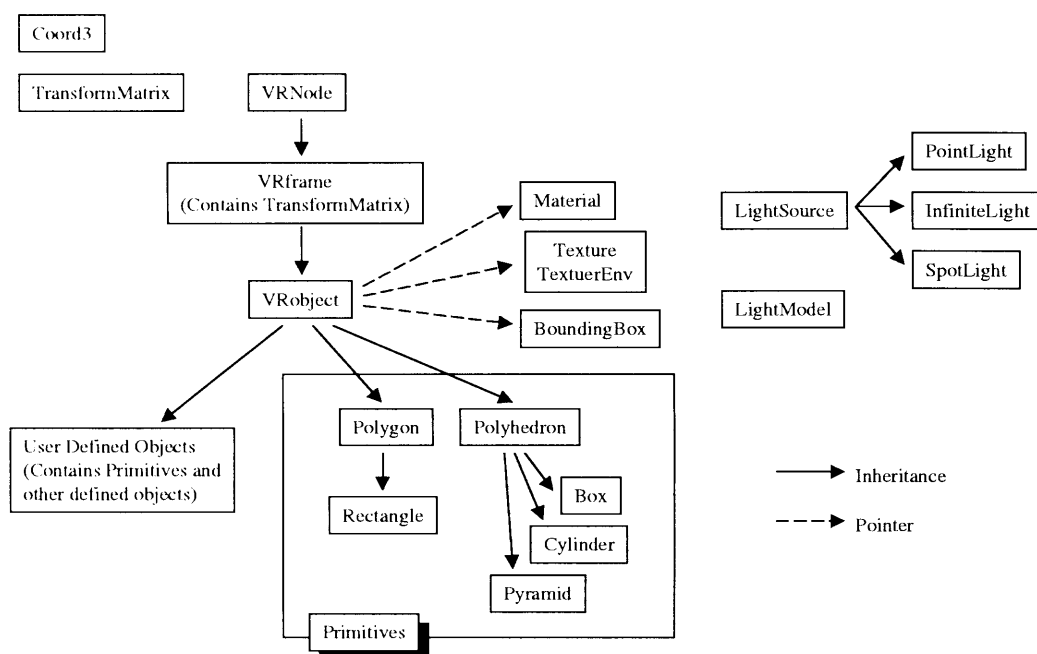


図 2.9 クラスライブラリの構成

この概念を図 2.8 に示す．すなわち，表示される物体から視点が遠い場合は，そのモデルは小さくしか描画されないので精密なモデルを使用する必要がなく，粗い簡略化したモデルを用いて描画することができる．逆に，物体が近距離にある場合は詳細な部分まで見えてしまうので，モデルそのものを詳細にしておく必要が生じる．Level Of Detail は，必要な詳細度を判定して描画に使用するモデルを動的に切り替える仕組みである．

オープンな環境のウォークスルーやドライブスルーなどでは，遠くにある物体はそれだけ数多く視野内に入ってくるため，Level Of Detail 制御を用いることにより描画される物体の総ポリゴン数がある程度一定に保ち，その結果一定のフレームレートを維持するという効果が期待される．

2.4.4 ライブラリの実装

本システムで開発したライブラリは，当初 Silicon Graphics 社のワークステーション上で動作する三次元グラフィックスライブラリ IRIS GL を用いて実装された．IRIS GL は OpenGL のベースとなったライブラリであり，三次元グラフィックスの分野では非常によく利用されてきた．しかしながら，三次元グラフィックスの需要が拡大するにつれ，異機種間で互換性のあるソースコードの必要性が増してきたため，特定の機種だけでなく PC も含めた広範囲なプラットフォーム上で動作する互換性の高いライブラリが求められるようになった．このような背景を受けて，IRIS GL をベースに機種依存性を排除した新たなグラフィックスライブラリとして制定されたのが OpenGL である．OpenGL はパーソナルコンピュータ上で動作する Microsoft Windows 95/98, Windows NT/2000 上でも動作し，さらに PC UNIX の代表格である Linux にも移植されており，最も機種を問わないグラフィックスライブラリとして定着している．本ライブラリでも，このような動向に従い，OpenGL ベースに移行した．

図 2.9 に，開発したクラスライブラリの構造を示す．現在，クラスライブラリには以下に述べる種類のクラスを用意している．これらは座標系クラス，ツリー構造クラス，基本物体クラス，マテリア

ル関連クラス，光源計算関連クラスに大別される．

座標系クラス

空間中の点と座標系の位置関係を表現するためのクラス．Open GL では，整数変数や浮動小数点変数の配列へのポインタを用いてグラフィックスライブラリとデータのやりとりを行うようになっている．三次元空間の操作を行うにあたって，配列の形式でデータを扱うのはいささか不便であり，三次元座標などをひとまとまりにして通常の変数と同じように扱える方が便利である．そこで，三次元空間上の点や座標変換を表すデータをひとまとめにすると共に，それら进行操作するインタフェースを定義して，クラスとして実装した．

- Coord3: 3次元座標および3次元ベクトルを表すクラス．これは x, y, z 座標を表す3つの浮動小数点数の組であり，

$$p = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (2.15)$$

という 4×1 のベクトルで表現される．最下行の 1 は，後述する同次変換行列との演算表記をわかりやすくするためのものであり，実際の演算には利用されない．この三次元座標を部品として定義することにより，これら3つの数をまとめて扱うことを可能にする．また，三次元座標の絶対値，実数倍，三次元座標同士の加減算，内積，ベクトル積などを定義し，座標系の操作を容易にする．実数倍，加減算については $*$, $+$, $-$ の演算子記号を用いて直観的な表記を行なう．例えば，プログラム中において

a = b + c; (a, b, c は三次元ベクトル)
a = b - c; (a, b, c は三次元ベクトル)
a = r * b; (a, b は三次元ベクトル, r は実数)

という表記が可能になる．このような表記法は C++ 言語における演算子の多重定義機能を用いて実現されている．また，Coord3 クラスは座標系のみならず赤 (R)，緑 (G)，青 (B) からなる色情報を表現するためにも使用可能である．新規に生成された Coord3 クラスのオブジェクトは，ゼロに初期化される．

- Coord4: グラフィックス生成に用いられる4個の数の組を表すクラス．色情報において，R, G, B の他に透明度を加えた R, G, B, A の組を扱う場合や，三次元座標に「拡大・縮小率」w を加えた組を扱う場合に用いる．

$$p = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \quad \text{または} \quad p = \begin{bmatrix} r \\ g \\ b \\ a \end{bmatrix} \quad (2.16)$$

という 4×1 のベクトルで表現され，Coord3 クラスと同様の演算子を使用できる．

- TransformMatrix: 同次座標変換行列 (4×4) を表すクラス．座標変換行列は，三次元空間内のそれぞれの物体に固定された座標系同士の関係を記述するもので， 4×4 の同次変換行列で表

現する．

$$T = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} = \begin{bmatrix} a & e & i & m \\ b & f & j & n \\ c & g & k & o \\ d & h & l & p \end{bmatrix} \quad (2.17)$$

ここで、 a, b, c, \dots は行列データ格納の順番を示すものであり、OpenGL における同次変換行列の順序に従っている．すなわち、データは列方向に並んでおり、行列要素の表記 m_{ij} を素直に C 言語にあてはめた場合と逆になっている．これは、前身の IRIS GL が「行ベクトル表記」であったことの名残であると考えられるが、本ライブラリではデータアクセスのためのインタフェースを通して操作を行うので、プログラマにとってはさほど重要な問題ではない．OpenGL のデータ順序にあわせたのは、OpenGL が用意するシステム行列とのやりとりを行う際の効率を考慮してのことである．

TransformMatrix クラスでは、この座標変換行列に対する平行移動、スケール変換、x, y, z 軸および任意の軸の回りの回転、xy, yz, zx 平面および任意の面に関する反転、逆行列、同次変換行列同士の乗算を定義し、扱いを容易にしている．また、三次元座標との乗算（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．

これらの演算のうち、平行移動、回転、スケール変換、乗算は内部的に OpenGL の行列演算関数を利用することも可能である．逆行列に関しては、現在の行列が回転と平行移動のみで構成されている場合、すなわち \mathbf{R} を姿勢をあらわす 3×3 行列、 \mathbf{p} を位置を表す 3×1 ベクトルとして

$${}^A_B T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & x \\ r_{21} & r_{22} & r_{23} & y \\ r_{31} & r_{32} & r_{33} & z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^A_B R & {}^A P_{Borg} \\ \mathbf{0} & 1 \end{bmatrix} \quad (2.18)$$

という形で表現されている場合は、行列の逆行列は回転行列部分の転置と部分的な乗算により実現され、計算量の多い逆行列計算を行わなくて済む．そこで、平行移動と回転以外の操作が施されていない行列に対する逆行列計算はこの方法により行うように実装した．

シーングラフ基本構造

シーングラフのツリー構造自体は、横方向の単純なリストと縦方向の親子関係により表現される．

- VRList: VR 世界を構成するツリー構造において、子オブジェクトを管理するための 2 重連結リストのヘッダを表すクラス．
- VRNode: VR 世界を構成するツリー構造のノードを表すクラス．親ノードへのポインタ、同じリストに属する前後のノードへのポインタ、および子ノードのリストのヘッダを持つ．

これらのクラスを利用してシーングラフの階層構造を実現する仕組みを図 2.10 に示す．

VR 物体の基本クラス

- VRframe: 「フレーム」（座標系）を表すクラスで、VRNode クラスを継承している．このため、世界を構成するツリー構造を保持するノード情報を含み（図 2.10）、子物体の登録・削除を行なえる．このクラスのインスタンスには以下の 3 種類の TransformMatrix クラスのオブジェクト（座標変換行列）が含まれる．

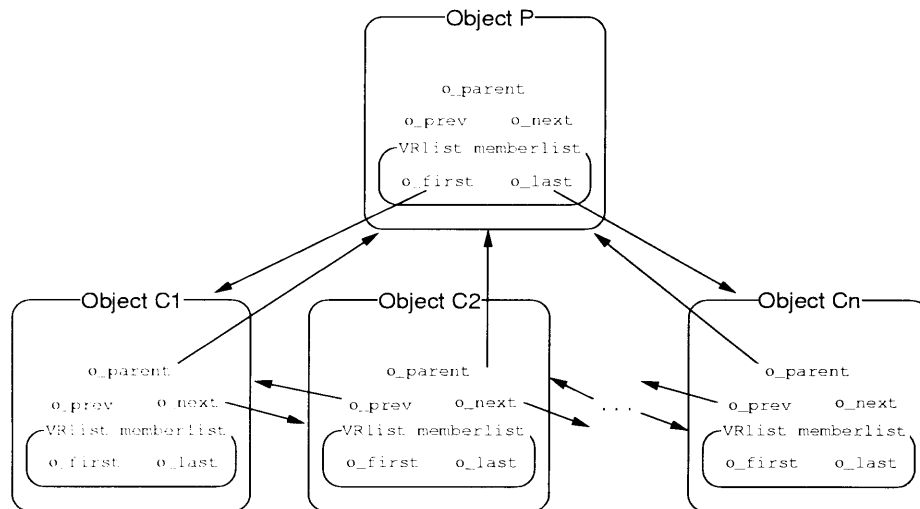


図 2.10 シーングラフの階層構造

1. 基準座標系（「親」物体に付随する座標系）に対するその座標系の相対的な座標変換（ローカル行列），通常の平行移動，回転などの座標変換操作はこの行列に対して行なわれる．すなわち，通常の座標変換操作は親物体の座標系を基準としてローカルに行なわれる．
2. 幾何コンパイル（2.4.3 節）のルートに対するその座標系の変換（幾何コンパイル行列）．親物体との間で幾何コンパイル計算を行うかどうかの設定，幾何コンパイルの実行・解除を行なうことができる．
3. 世界基準座標系に対するその座標系の変換（グローバル行列）．世界基準座標系に対する変換行列の取得，位置および変換行列全体の設定が行なえる．この行列に対する設定を行なった場合，ツリー構造に基づき親物体の変換行列を参照してローカル行列の設定も自動的に行なわれる．

このクラスのインスタンスに対しては，ローカル行列の設定（代入），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．

- **VRObjcet**: 「汎用バーチャルオブジェクト」を表すクラスで，バーチャル環境におけるすべての物体の基本となるクラスである．このクラスは **VRframe** クラスを継承しており，**VRframe** クラスに加えて，描画処理に付随する変換行列計算の管理，マテリアル属性およびグラフィックスシステムに対するマテリアル指定の管理，テキストチャ指定，オブジェクト・カリング機能を備えたものである．描画を行なうにはこのクラスのオブジェクトの **draw()** という関数をコールする．そのオブジェクトに子オブジェクトが設定されていれば，ツリー構造に従って子オブジェクトの描画も行なう．また，このクラスは表面材質 (**Material**) クラスへのポインタを保持しており，物体のマテリアル属性を指定することができる．描画の際には指定したマテリアル属性で描画される．同様に，**Texture** クラスへのポインタを持ち，物体表面のテキストチャを指定でき，描画の際には指定したテキストチャを貼り付けて描画が行われる．
- **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．バーチャル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われる．

「汎用バーチャルオブジェクト」クラスを継承して、基本的な物体（プリミティブ）が用意されている。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。

- Polyhedron: 一般的な多面体を表すクラスで、VObject クラスを継承して定義されている。Polyhedron クラスのオブジェクトは頂点座標を持ち、それぞれの面やエッジがどの頂点を含むかを示すインデックスを持っている。頂点、面、エッジの数の指定、頂点座標、面の法線ベクトル、各インデックスの設定を行なうインタフェースが用意されている。また、Polyhedron クラスを継承して、基本的な物体（プリミティブ）が用意されている。現在、この物体プリミティブとしては直方体 Box、円柱 Cylinder および四角錐 Pyramid のみ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リユーザ自身がプリミティブを設計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。
- Polygon: 一枚の面を表すクラスで、VObject クラスを継承して定義されている。このような面は体積を持たないので厳密には物理世界に対応した物体とは言えないが、簡略化したモデルを用いてバーチャル世界の描画を高速に行なうために用意されている。このクラスを継承して、長方形 Rectangle が定義されている。

実際の物体クラスは、「汎用バーチャルオブジェクト」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。

バーチャル環境の描画は、視点位置をトレイグ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。

マテリアル関連

マテリアルは、物体表面の材質、すなわち色や反射特性を定義するものである。また、テクスチャはさらに細かい質感を画像を貼り付けることにより表現するものである。ともに、物体表面の見え方を決定する要素である。

- Material: 「マテリアル属性」を表すクラス。透明度、アンビエント光係数、拡散反射係数、鏡面反射係数、放射係数等を設定することができる。こうして定義したマテリアル属性の特性は VObject クラスのオブジェクトに付与される。
- Texture: 物体表面に貼り付けるテクスチャを表すクラス。Silicon Graphics RGB フォーマットのファイルを指定し、物体表面全体に貼り付けることが可能である。VObject, Polyhedron およびその他のプリミティブクラスのオブジェクトにおいてテクスチャを指定することが可能である。

光源関連

- LightSource: 「光源」を表すクラス。このクラスは抽象クラスで、以下のクラスの基底クラスである。アンビエント光、色、位置等を指定することができる。また、光源を一時的に無効

にしたり，再び有効にしたりすることも可能である．

- PointLight 「点光源」を表すクラス．
- InfiniteLight 「平行光源」を表すクラス．
- SpotLight 「スポットライト光源」を表すクラス．照らす方向や，減衰係数，放射角度等を指定することができる．
- LightModel 「光源計算モデル」を表すクラス．シーン全体に渡るアンビエント光を指定したり，光源と面との間の距離による光の減衰効果をシミュレーションできる．鏡面反射の計算において，観察者の位置による影響を考慮に入れることもできる．さらに，面の両側において正しく光源計算を行なうようにすることも可能である．また，光源計算を行なわないようにすることが可能である．

2.5 アプリケーションプログラムの枠組み

本節では，ソフトウェアの技術的ポイントのうちクラスライブラリの中に組み込まずにアプリケーションプログラムで実現した事項について記述する．

2.5.1 分散処理とネットワーク対応

分散処理の目的

ハードウェアの節で記述したように，本システムでは運動計測と世界管理・描画を独立のコンピュータに担当させている．これは，それぞれの仕事が互いに待たされることなく円滑に実行されるための構成である．運動計測用コンピュータ（NEC PC-9801）では MS-DOS 上の計測プログラムが動いており，運動状態を数値化して共有メモリに書き込む．PC-9801 と共有メモリインタフェースで接続された SGI Indigo2 ワークステーションにおいて VR 空間の管理・更新と画像生成の仕事を行う．この，運動計測とそれ以降の役割分担以外にも，本システムでは複数プロセスによる分散構成を採用している．

ある程度以上大規模なモデルをグラフィックス表示することは非常に負荷の大きい仕事であり，画像表示を行うコンピュータがどれだけこの仕事に専念できるかによってグラフィックスの更新レートが影響される．グラフィックスの更新レートはユーザの操作に対する応答特性を直接決定付けるものであるから，世界モデルの更新などの画像生成以外の仕事はなるべく他の計算機に行わせ処理を分散させた方がシステム全体の応答特性が向上することが期待される．また，複数の操作者が同じ空間を共有し議論を行いながら同一あるいは異なる視点から空間探索を行う場合など，複数の画像出力を行う必要が生じ，この場合にも空間の整合性を保ちつつ別々の計算機で画像生成を行うことが有効である．本システムにおいてはマルチユーザ構成を想定していないが，初期のハードウェア構成において HMD へ出力する左右用の画像を別々のコンピュータで生成する方法をとったため，左右それぞれの画像を生成する描画プロセスを分離する必要があった．

ネットワークを介した情報の共有には，共有すべき情報の質と量，応答性，ネットワークの規模，通信回線の速度などに応じて，いくつかの段階が存在する．その両極端は，次のような形態となる．

1. 密な情報共有：比較的近い距離の計算機間で，高速な LAN などを通じて情報共有を行うレベル．細かい情報のやりとりや，実時間あるいは頻度の高い情報更新に適する．

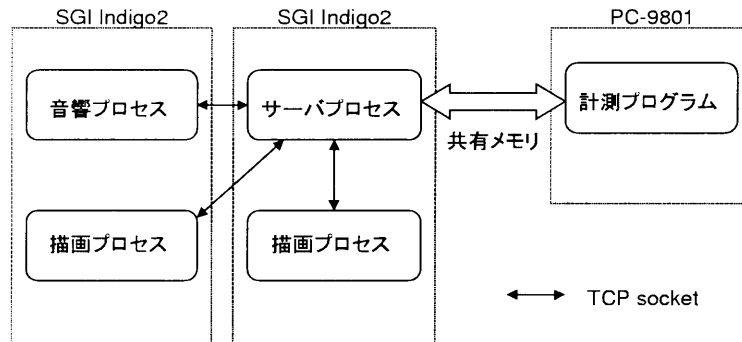


図 2.11 プロセス構成

2. 疎な情報共有：インターネットを介した世界的規模での情報共有など，比較的遠い計算機間で共有を行うレベル．World Wide Web などの利用形態がこれにあたる．通信する地点の間を Point-to-point で結ぶなどの手段を講じない限り，一般的な Internet では実時間性を確保することは難しい．

本システムでは，遠隔地の間で共有 VR 環境を構築することを目的としている訳ではなく，単一サイトにおけるシステム構成の柔軟性確保と負荷分散をネットワーク対応の主な目的とするため，「密な情報共有」のための通信手段を提供する．すなわち，各々の役割を受け持つ計算機上にプロセスを走らせ，プロセス間通信によりデータの整合性を保ちつつ分散処理を行うものである．プロセス間通信をネットワークを通じて行うように設計しておけば，それらのプロセスが別々の計算機上で動作しているかあるいは同一の計算機上で動作しているかに関係なくシステムを構築できる．

プロセス構成

本システムにおける役割分担は，以下の通りである．

- データサーバ：世界更新・複数のグラフィックス画像生成などを整合性を保ちつつ行うには，一元的にデータを保持し管理することが有効である．このデータの管理を行うプロセスをサーバプロセスと呼び，サーバが他の役割を持つクライアントプロセスと通信を行って全体の整合性を保持する．
- グラフィックス画像生成：VR 空間画像の生成を実行する部分．システムの構成に応じて，左右両眼の画像生成のために別々のプロセスを割り当てたり，一つのプロセスで左右両眼の画像を生成する形式が考えられる．
- VR 世界計算：物体の挙動などの VR 世界更新のための計算負荷が大きい場合は，この計算のためのプロセスを分離し独立の計算機で動作させることが有効であると考えられる．負荷が小さい場合はサーバプロセス内で計算を行わせてもよいが，分離することにより世界計算をバックグラウンドで行い，更新計算のための待ち時間を小さくして全体の応答特性を向上させる効果が期待できる．

後述するように，本システムで構築したアプリケーションでは大部分が建物などの固定物体であるため，実際には世界計算を分離せずサーバプログラムの中で行っている．システム全体のプロセス構

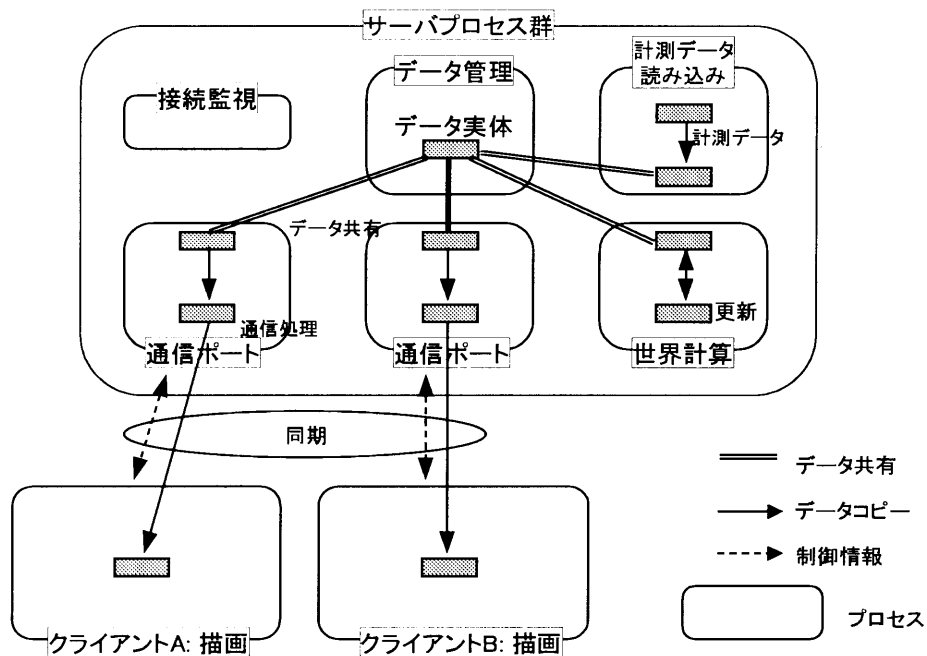


図 2.12 サーバプロセスグループ

成を図 2.11 に示す。サーバプロセスは VR 空間の一元的管理を行うものであり、共有メモリから運動情報を読み出して VR 空間の更新と描画プロセスへのデータの配信を行う。描画プロセスはサーバプロセスと同一のマシン上で実行してもよい。描画プロセスは 1 回の処理（1 画面生成）に時間のかかるタスクであるため、サーバプロセスと描画プロセスの間の通信はキロヘルツオーダーの時間精度が要求される訳ではなく、サーバプロセスと描画プロセス間の通信は汎用性の高い TCP/IP を用いることとした。

サーバプロセスは、それ自体複数のプロセスから構成されるプロセスグループであり、グループを構成する各サブプロセスはそれぞれが専門の仕事を行う（図 2.12）。

- 接続待ち：クライアントからの接続を常に監視し、接続要求を受けると「通信ポート」サブプロセスを生成する。
- 通信ポート：クライアントとの接続が確立された後、実際の通信を継続的に行う。1 つのクライアントプロセスには 1 つの通信ポートが対応する。
- 運動情報の読みとり：初期の構成では、運動情報をシリアル通信 (RS-232C) によりワークステーションへ転送していた。一組の運動情報を読み込むまでに時間がかかり、他の仕事をブロックしないために分離した。後期のシステムでは共有メモリを採用することにより変数のアクセスだけで運動情報の読みとりができるようになったため、読みとりは世界更新プロセスの中で行っている。
- 世界更新：接続されているクライアントへ供給するデータを更新するプロセス。

これらの役割にそれぞれプロセスを割り当てることにより、複数クライアントとの通信とデータ管理・更新の見かけ上の同時実行をオペレーティングシステムが提供するマルチタスク機能に委任できる。

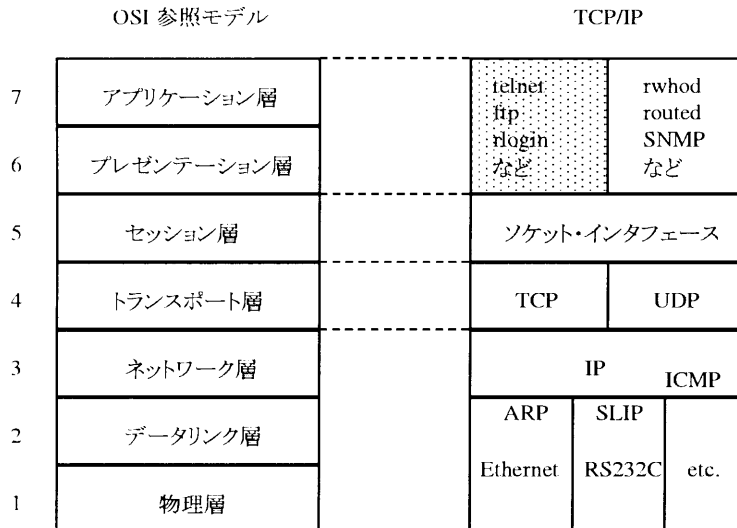


図 2.13 ネットワーク通信のプロトコルスタック

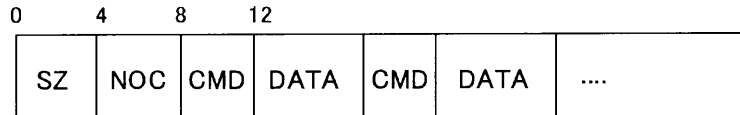
運動情報および VR 空間の状態に関する情報はプロセスグループ内において唯一の実体を持つプライマリデータによって管理され、各サブプロセスはこれらのデータを共有することにより全体としてのデータ整合性が確保される。サブプロセスは適宜プライマリデータを参照し、参照中はプライマリに対する排他制御が行われる。サブプロセスが受け持ちの仕事の間プライマリデータをブロックしないよう、必要に応じてサブプロセス内にデータのコピーを作成し、プライマリデータ自体は短時間で解放するように設計している。

データの整合性を考える場合、もう一つ重要な要素は時間的な同期である。複数のクライアントで単一の実体を持つデータからコピーされたデータを参照したとしても、参照の時間がずれてその間に別のクライアントとの通信により大元のデータが更新されてしまったのでは、データの整合性が確保されているとはいえない。この矛盾を解消するため、サーバプロセス群の間で通信のタイミングの同期をとる仕組みを導入する。例えば、HMD やマルチスクリーンシステムにおいて複数画面の画像生成を別々の描画クライアントが担当する場合、それぞれの描画処理におけるダブルバッファの表裏切り替え（フレーム切り替え）のタイミングを同期させ、それぞれのフレームで必ず同一のデータを元に描画を行うようにすれば不自然な画像のずれは生じない。本システムにおいては、各描画クライアントが 1 画面描画し終わった後サーバに描画完了を知らせ、サーバがすべての描画クライアントから描画完了報告を受け取った時点で各描画クライアントにフレーム切り替えを許可し、次画面描画用にこの時点の共有データを配布することにより、同期を保つようにした。

通信プロトコル

近隣のコンピュータ同士で分散処理を行うための通信プロトコルとしては、できるだけ通信量を少なくし、サーバとクライアントの間の通信負荷を抑えるとともに高い更新レートを確保できるようにすることが望ましい。

ネットワーク通信の標準である TCP/IP プロトコルスタック（図 2.13：[68] より再構成）において、ユーザが自由に設計できるのはトランスポート層より上位のアプリケーション層である。実際には、OS に用意されているソケットインタフェースを用い、トランスポート層として次のどちらを利



SZ: パケットのサイズ(整数 4バイト)
 NOC: コマンドとデータの組の数(整数 4バイト)
 CMD: コマンド(整数 4バイト)
 DATA: データ本体(可変長)

図 2.14 通信に使用するデータ形式

用するかを選択する。

- TCP (Transmission Control Protocol): TCP は IP ネットワーク上の通信路を確保し (コネクション型), データの到達可能性, 順序性を保証する仕組みを持つ. このため, アプリケーションでは送られたデータが, 順番の入れ替わり, 損失, 重複などがなく確実に届くものと仮定して利用することができる. その反面, 下位層の信頼性が十分でない場合やネットワークが混雑している場合にはこうした信頼性確保のためのパケット再送が生じ, 余分なトラフィックを発生させることになる.
- UDP (User Datagram Protocol): TCP のようなコネクションを張らずに単純にデータを送りつける方式. 実際, IP データグラムにチェックサムとポート識別機能を加えただけのものである. TCP が行っているような信頼性確保が必要ない, あるいはアプリケーション側で独自に行う場合に利用する.

本システムにおいては, 近接したコンピュータ間の通信をターゲットとするため, 通信路の信頼性は高いと仮定する. このため, アプリケーション側でデータの整合性チェックが不要で, データの取りこぼしを気にする必要がない TCP を採用した. 同じネットワーク対応といっても, より遠隔地との通信を行う場合には, 異なる戦略が必要である.

通信データ形式に関しては, 通信量を最小に抑えサーバ・クライアント双方での処理を単純化するため, すべて単一形式のバイナリデータにより行うこととした. ネットワークアプリケーションで用いられる多くのプロトコルでは, 異機種間の互換性を重視するため接続初期のやりとりを ASCII 形式の文字データで行い, それに続くデータ本体をテキストまたはバイナリ形式でやりとりするが, 本ソフトウェアではあえて単一の形式により単純化し, 効率化を図る.

このバイナリデータ形式は, なるべく汎用性を持たせ, かつ単純で通信量を最小にとどめることを念頭において設計を行った. データの形式を, 図 2.14 に示す. 本データ形式は, 様々な種類のデータをまとめて送ることにより, 送受信を行うシステムコール (UNIX では `send()`, `recv()`) の回数を減らし通信のオーバーヘッドを少なくするとともに, サーバ・クライアント間でのやり取りの回数を抑え相手側からのデータ待ち時間を少なくする効果を狙っている. 本データ形式は, 基本的にはデータの種類もしくはコマンドの識別番号 (CMD) とデータ本体 (DATA) を一組として, これを複数個並べただけのいたって単純なものである. CMD の種類によっては DATA が存在しない (データ本体のサイズが 0) 場合もある. 一回で送受信するパケットは, これらの CMD + DATA の組の前に, 組の数 (NOC) とパケット全体のサイズ (SZ) を付加したものである.

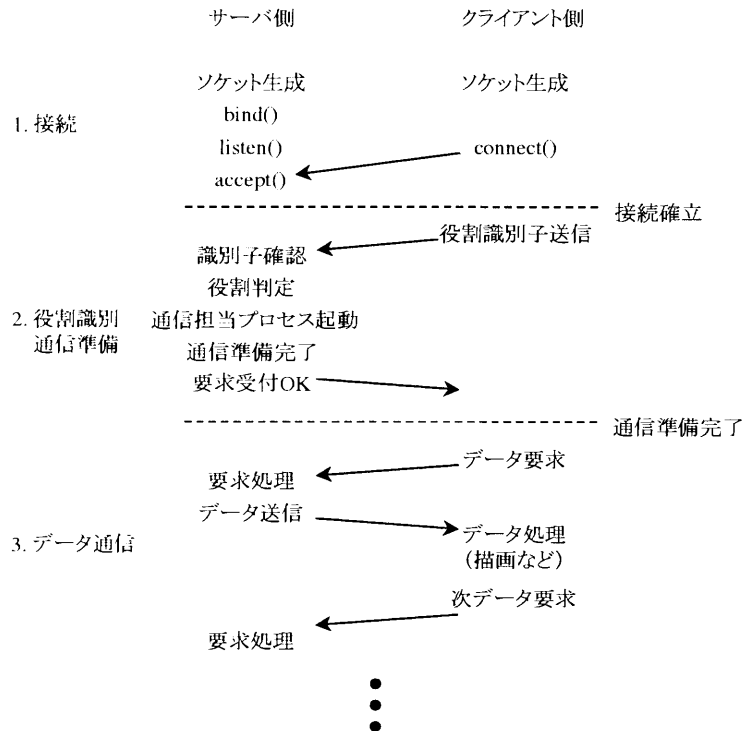


図 2.15 通信手順

データ送信側は、データのパック（梱包）を行う。すなわち、送りたいデータをその種類を示す識別番号と組にして送信バッファに並べ、送信バッファの先頭にパケット全体のサイズと組の数をセットした後、一回の `send()` システムコールで送信を行う。データ受信側は、データのアンパック（解梱）を行う。まず先頭の SZ の分の 4 バイトを `recv()` システムコールにより読み込み、続いて SZ の値のバイト数だけ受信バッファに読み込む。次に、受信バッファに読み込まれたデータの解析と処理を行う。

1. 整数 4 バイト分の「コマンド/データの数」(NOC) を読み取る。
2. バッファの先頭側から整数 4 バイト分のコマンド/データ種類識別番号 (CMD) を見て、その内容から処理すべき仕事を判断する。
3. それぞれの CMD 番号に対応してデータ本体 (DATA) のサイズが決められており、バッファ内で CMD に続くこのサイズの内容を DATA と見なし、その値を用いて処理が行われる。
4. CMD と DATA のサイズの分だけポインタを進める。2 から 4 を NOC 回繰り返す。

最後に、通信手順に関して記述する。まずクライアントがサーバに接続を行った時、クライアントプロセスが行う仕事の役割をサーバ側で識別する必要がある。サーバはその役割によってクライアントと送受信するデータの種類を決定する。この手順は、以下のように行われる。

1. クライアントからサーバへ接続要求を行う。このとき、サーバプロセスは既に起動しているものとする。サーバプロセスが見当たらない場合、クライアントは接続エラーを起こして終了する。接続要求の手続きは、通常の UNIX のサーバ・クライアント形式の手順で行う。

2. 接続が確立されると、まずクライアントからサーバへ役割識別のデータを送る。このデータパケットは、NOIが1、CMDは「役割識別子」、DATAはクライアントの役割によって一意に定まる「役割番号」である。「役割番号」の値は、接続してきたプロセスが描画クライアントか、世界計算クライアントかによって異なる。サーバ側では、CMDの値が「役割識別子」以外の場合は間違っているあるいは対応していないクライアント種類の接続要求とみなして接続を切断する。
3. 以後、クライアントの役割に応じた内容の通信が行われる。クライアントは、一連の処理単位ごとに送受信を繰り返す。この間、接続開始・終了処理によるオーバーヘッドを避けるため、接続は保たれたままである。
4. クライアント・サーバのどちらかが相手へ終了信号を送ったり、あるいは何らかの理由により不慮の終了をした場合には、回線を切断する。

2.5.2 ハラメータ制御

作成したデータサーバプログラムでは、提示に関する各種パラメータを調整できるようにした。これらのパラメータを操作するため、グラフィカルユーザインタフェースを作成した。

アプリケーションの構成

サーバプログラムを起動すると、図2.16のGUIウィンドウが現れる。主な操作は、この中のControlメニューから行う。各メニュー項目はVR空間のモデルに依存するものと独立なものが混在しているが、以下簡単に解説する。

- Existence Mode: 視点をバーチャルオペレータあるいはバーチャルロボットのどちらに設定するか切り替え。
- Locomotion Mode: 移動指令の前後・左右を体幹座標系と平行にするか、頭部座標系と平行にするか、あるいは無効にするかの切り替え。
- Locomotion Speed: 移動速度の調整ウィンドウ表示。
- Objects Display: VR空間に表示する物体の表示/非表示切り替え。
- Kinematics & Projection: 頭部平行移動・回転および手先位置の運動倍率指定と、視覚パラメータ（眼間距離、視野角）調整ウィンドウ（図2.17 (a)）表示。
- Time Delay: 時間遅れ調整ウィンドウ（図2.17 (b)）表示。
- Light Source: CG画像生成における光源（点光源/平行光源）の切り替え。
- Rendering Model: サーフェス表示/ワイヤフレーム表示の切り替え。
- Home Position: 視点位置を初期位置に戻す。
- World Initialize: VR空間中の可動物体を初期状態に戻す。

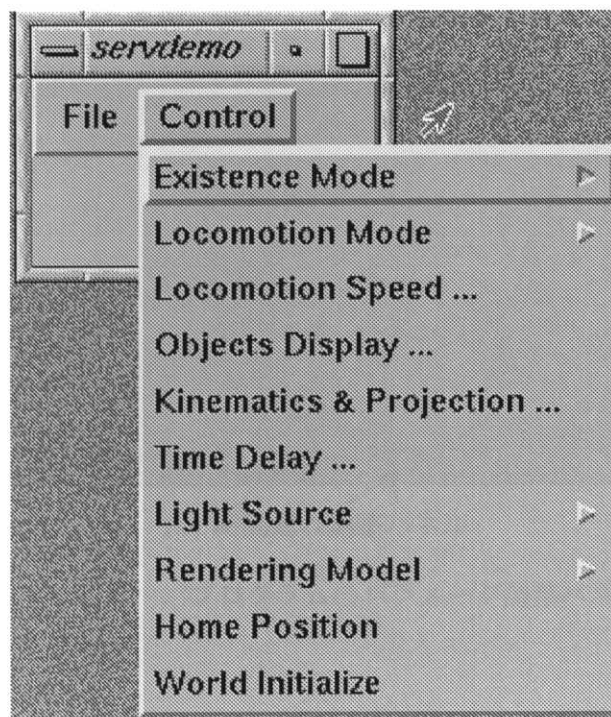
これらのうち、Kinematics & Projectionのパネルは第3章における議論の検証実験において利用する。ここでは、VR環境のバーチャル人間における眼間距離、視野角、物理視野に対する有効視野（周囲をブラックアウト）、頭部平行移動倍率、頭部回転運動倍率、手先位置の運動倍率を指定することができる。



(a) メインウィンドウ

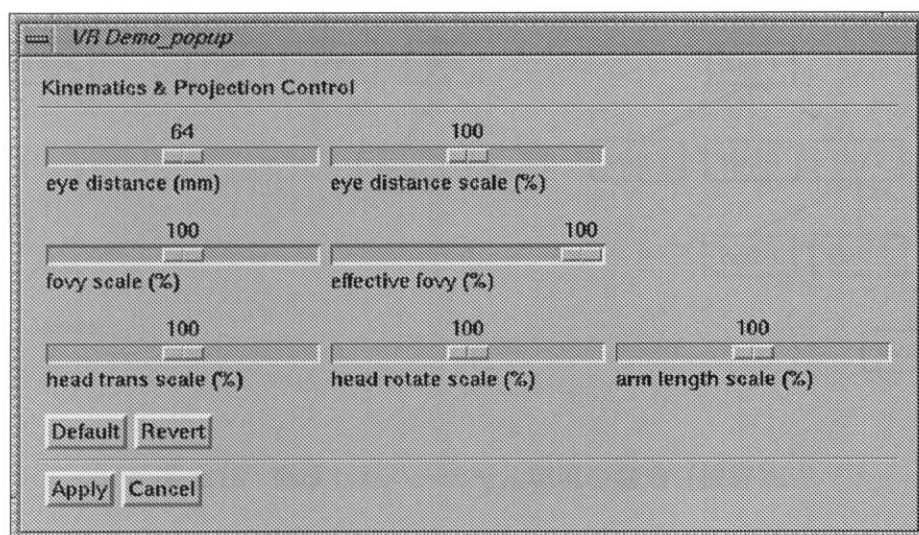


(b) File メニュー

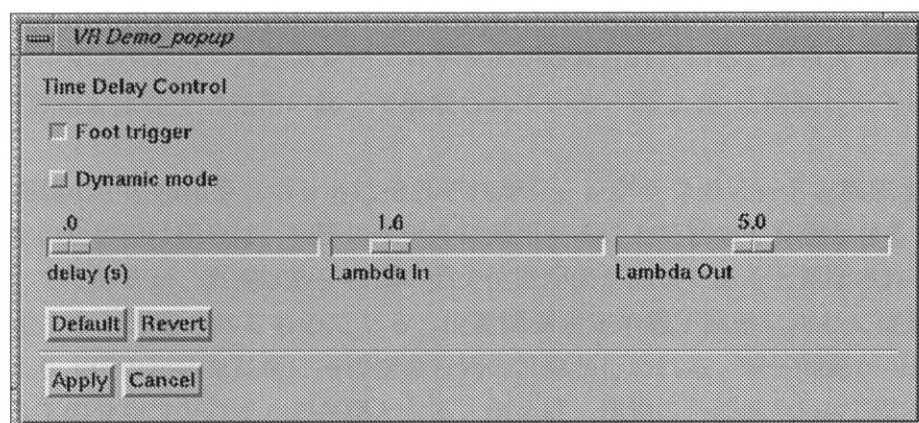


(c) Control メニュー

図 2.16 サーバプログラムの GUI ウィンドウとメニュー



(a) 運動・視覚パラメータ調整パネル



(b) 時間遅れ調整パネル

図 2.17 サーバプログラムのパラメータ調整パネル

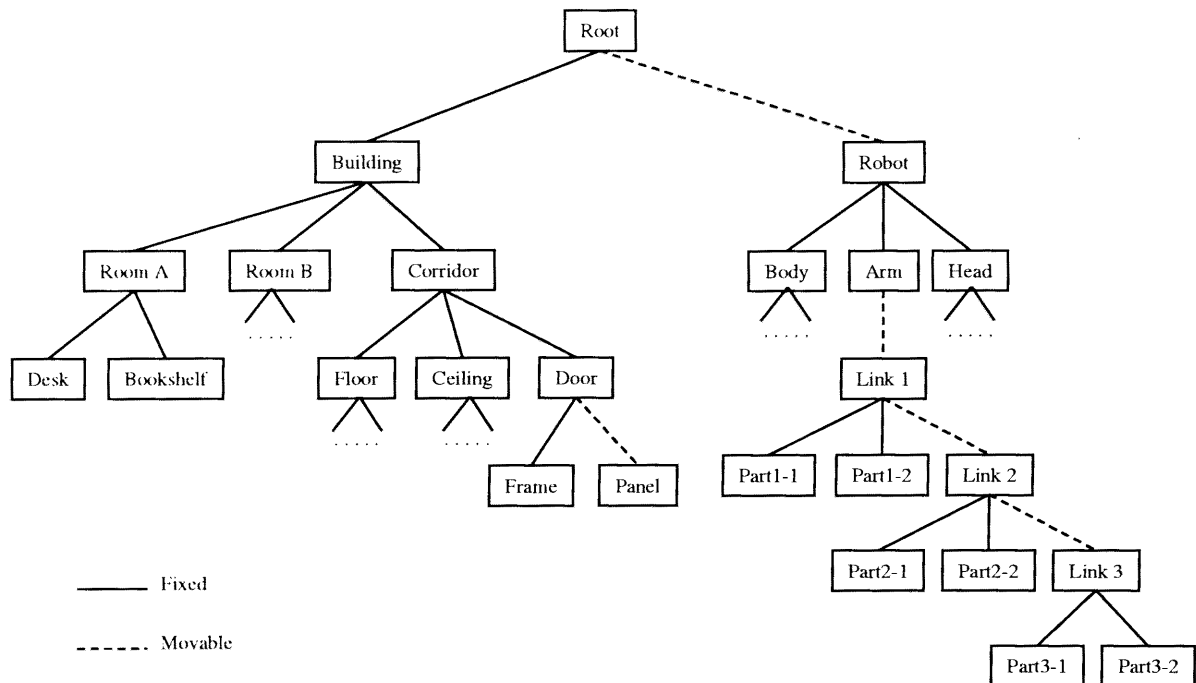


図 2.18 構成したバーチャル環境の構造（抜粋）。

2.6 VR 空間のモデリング

トレイグジスタンス型マスタ・スレーブシステムをバーチャル環境中に構築した。環境はト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，東京大学駒場第2キャンパス 14 号館の実験室およびその建物の廊下で構成されている。図 2.18 に構成した VR 空間の構造を示す。

建物および部屋の中の物体やロボットはそれぞれがシーングラフのノードを構成する。2.4.3 節で説明した高速化のための仕組みは，実際のモデルに適用することで初めて有効となる。まず，幾何コンパイルについてであるが，実線は親子物体の間で位置関係が固定であると見なされるもの，破線は位置関係が変動すると想定されるものである。位置関係固定の親子関係に対しては幾何コンパイルが実行され，より上位の物体を基準に座標変換行列の計算が行われる。親子関係が可動の場合は，子物体がそれより下の幾何コンパイルの基準となる。例えば，建物の各部屋や柱は固定であるため，世界座標系を基準として幾何コンパイルを行うとジオメトリ計算の量を抑えることができる。一方，ロボットの各部分（アームを構成するリンクなど）およびロボット全体は可動であるため，これら可動部の要素をそれより上位の物体を基準にして幾何コンパイルを行うことは意味がない。このため，可動部において親子物体間の幾何コンパイルの「つながり」を切り，それぞれの部分内においては幾何コンパイルが有効となるように構成している。

次に，オブジェクト・カリングに関してであるが，空間は建物の各部屋，窓，ドア，あるいはロボットといった単位でオブジェクト・カリング判定を行うよう設定しており，シーングラフのトラバース時に現在の視点位置・方向を参照して描画実行・スキップの判定を行っている。特に，建物に関しては，部屋の中にバーチャル人間が存在する時や曲がった廊下に存在する時は方向として前方にある物体が手前側の壁に遮られて見えないため，汎用の視円錐と包絡球による判定ではなく独自の判定基準を組み込み，より効率的な描画処理を実現している。

さらに、Level Of Detail に関しては、建物の廊下の端に立って廊下全体を眺めたときに同時に視野内に入る描画対象のポリゴン数が最も多くなったことから、廊下のドアに LOD を導入してポリゴン数削減を図った。詳細なモデルではドアパネルの凹凸が存在しドアノブがポリゴン数の多い円柱で表現されており 136 ポリゴンで構成されるが、簡略化モデルではドアパネルを 1 枚の板で表現しノブを立方体に簡略化することで、12 ポリゴンに留めている。廊下全体でドアは約 20（そのうち両開きが半分ほどあるのでパネル数は約 30）ほど存在し、全部に詳細なモデルを使用するとこれだけで 3000 ポリゴンほどになるが、視点からの距離が 5m 以遠のドアは簡略化モデルを用いて描画することで、大幅なフレームレートの向上を図っている。

作成したバーチャル環境は、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と配置を正確に反映している。これにより、自然な 3 次元空間が構成される。

本プログラムでは、自分がバーチャルオペレータとしてバーチャル環境内に存在するモードと、バーチャルロボットの中に入り込んでト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、バーチャルオペレータへのトレイグジスタンスとバーチャルロボットへのトレイグジスタンスが行なえる。バーチャルオペレータの運動とバーチャルスレーブロボットの動きはト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バーチャル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。

バーチャルオペレータモードの時は、手を伸ばすとバーチャル人間の腕や手が見え、下を向くと足や体が観察できる。このバーチャル人間はバーチャルマスタシステムのオペレータ席に座っており、バーチャル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。バーチャルロボットモードの時は、実環境でのト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、バーチャル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、バーチャル環境への自己投射性が実現されている。

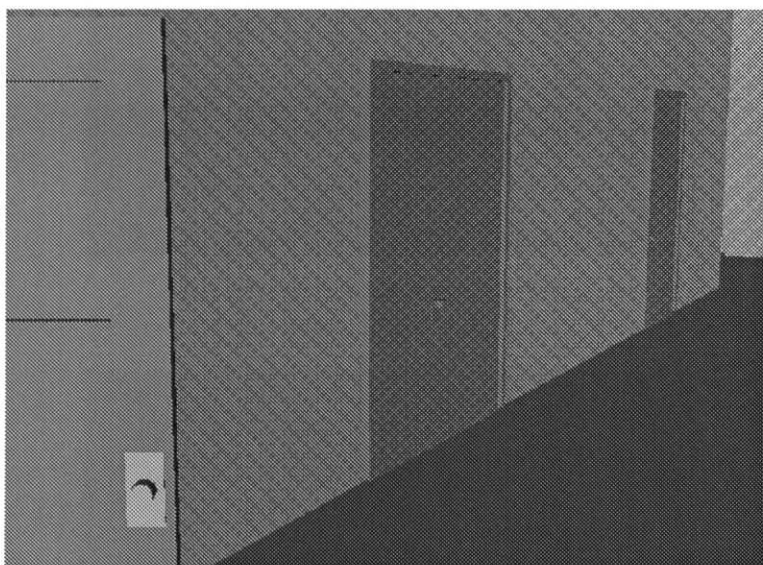
バーチャルオペレータとバーチャルスレーブロボットは、ジョイスティックの操作によってバーチャル環境内を自由に動き回ることができる。これは、オペレータがバーチャル空間内を自由に行動するという要件を実現したものである。

また、バーチャル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バーチャル環境内に実現した。これはオペレータとバーチャル環境との相互作用の実現例である。これらの作業は、バーチャルオペレータ自身が行なうことも、バーチャルスレーブロボットにト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。バーチャル環境の画像の例を図 2.20 に示す。

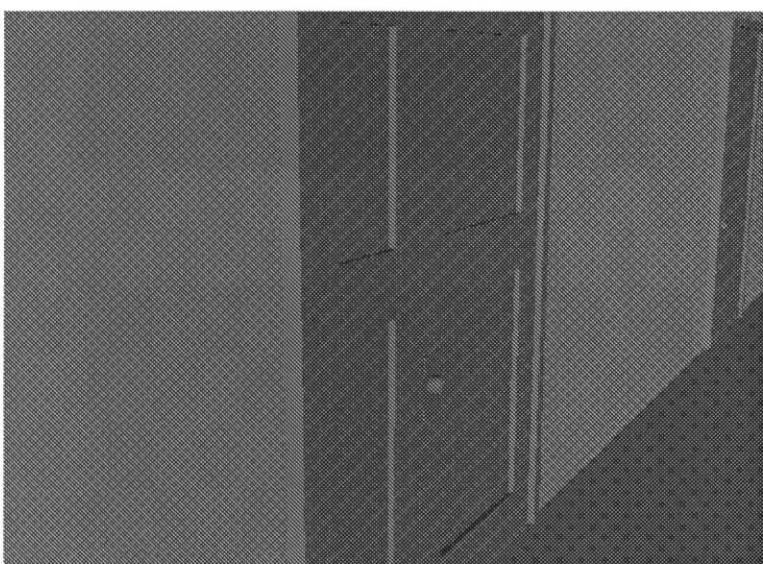
バーチャル環境中に存在する各物体は、次のように構成されている。

- 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：136ポリゴン（簡略化モデルは 12 ポリゴン）
- 窓：30ポリゴン

構築されたバーチャル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実条件に描画すると実時間での制御は難しいが、2.4.3 節で述べた幾何コンパイルおよびオブジェクト・カリングの利用と視点からの距離に応じた Level Of Detail 制御による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。Level Of Detail 制御は、物体が描画される時点で視点からの距離を検知し、モデルを選択することにより表示ポリゴン数を調整している。

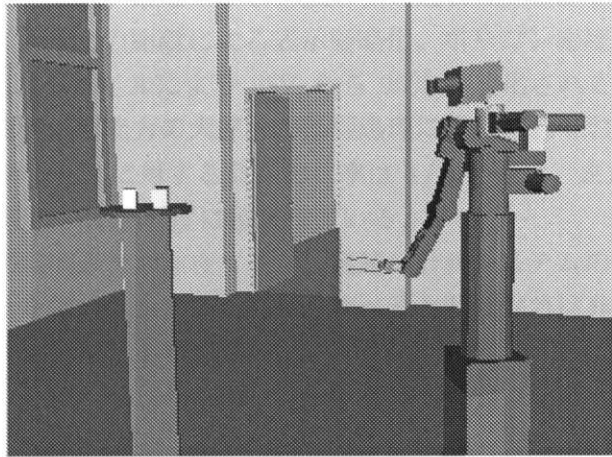


(a) 遠方から見たドア：パネルが単純な面で構成される．

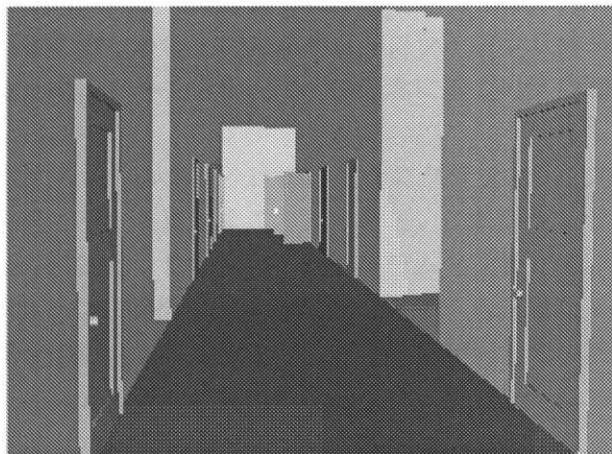


(b) 近距離から見たドア：パネルの凹凸が存在する．

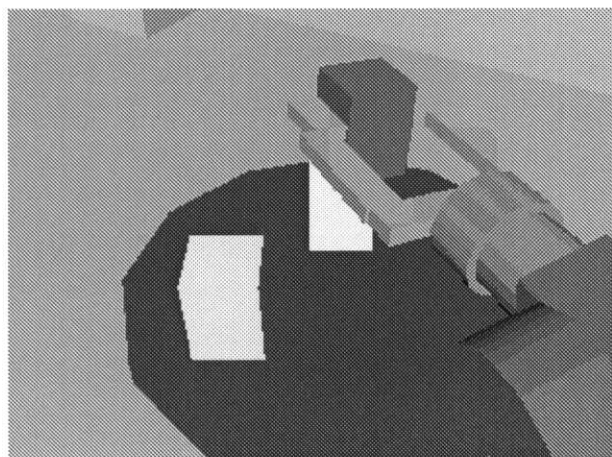
図 2.19 LOD 機能を利用したモデル構築例



(a) バーチャルスレーブロボット
(a) Virtual slave robot.



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



(c) バーチャル環境における積木作業
(c) Block building task in virtual environment.

図 2.20 バーチャル環境の画像例

全 19 自由度を制御し、これらの物体を画像表示して、20～60Hz の描画レートが得られている．2.4.3 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とオブジェクト・カリングにより 2～3 倍、さらに物体との距離による表示簡略化、ポリゴン密度の高いロボットなどのオブジェクト・カリングと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．

画像生成のために若干の時間遅れが存在するが、高速な計測システムの効果と高速化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．

2.7 まとめ

運動感覚と視覚が操作者において統合されるバーチャル環境へのトレイグジスタンスシステム構築を目指して、システム全体の時間遅延を最小化したシステムを構築した．また、VR 空間の系統的管理と視覚提示画像生成の高速化のための手法を組み込んだライブラリ体系を構築し、さらに VR 空間への自己投射性を実現するため、バーチャル人間への概念に基づいて空間のモデリングとアプリケーション構築を行った．

構築されたシステムにおけるボトルネックはほぼ完全にグラフィックス画像生成段階であり、運動計測による遅延は全体から見てゼロと見なすことができる．これはセンサシステム自体に無視できない遅延を持っていた、市販の磁気センサを利用したシステムと比較して非常に良好な応答性を実現したものであり、特にグラフィックス画像生成が短時間で完了する簡単な VR 空間を提示する場合には極めて良質な視覚と運動感覚の統合を実現する．この特性を利用して、次章では従来システム全体の遅延が存在したために定量的な分析が行われてこなかった、視覚提示における静的な立体視と遅延に依存しない動的要素との関係について言及する．

第 3 章

HMD 使用時の視覚系パラメータ整合性の影響

3.1 概要

トレイグジスタンスおよびバーチャルリアリティ (VR) 技術 [64] の特色は、人間が遠隔あるいはバーチャル環境に実際に存在しているような臨場感を与えることにより、人間の持つ高度な判断能力やスキルを有効に利用することにある。そのためには、操作者に対して遠隔あるいはバーチャル環境を自然な三次元空間として視覚的に認識させることは極めて重要である。

人間に対して視覚的に三次元空間の提示を行うための研究の歴史は古く、例えば立体テレビなどの実現に向けた三次元映像提示の研究開発が数多く行われている [41] [25] [10]。現在までに実用化されている立体ディスプレイは、その多くが二眼式（ステレオ画像方式）であり、これは人間の視覚による空間知覚手がかりのうち両眼輻輳角と視差を利用している。頭部搭載型視覚提示装置 (Head-Mounted Display: HMD) もその一種であり、1968 年に Sutherland により初めて開発 [57] されて以来、その後画像表示デバイスの進歩と小型化、さらに頭部の三次元的な位置・姿勢を手軽に計測できる磁気センサの商品化と相まって、1990 年頃から加速度的に普及している。HMD は VR システムの一つのスタイルを形作ったキーデバイスであり、そのパーソナルデバイスとしての特性ゆえに、特に空間コストの高い我が国においては今後も VR の主要なデバイスの一つであると考えられる。

HMD を用いて自然な三次元空間を生成するための研究は、遠隔実世界をビデオ画像を用いて提示する実環境へのトレイグジスタンス [58][60][31][67] およびバーチャル世界をコンピュータグラフィックス (CG) 画像により提示する狭義のバーチャルリアリティ [49][23][50][51] のそれぞれの観点から行われている。後者はバーチャル世界へのトレイグジスタンス [65][43][76] とも呼ばれる。操作者に自然な三次元空間を認識させるためには、無造作にステレオ画像を提示すればよい訳でなく、提示デバイスに対応した正確な視覚的空間知覚手がかりを操作者に与えることが必要不可欠である。すなわち、両眼に提示する画像をどのように生成するかは重要なポイントである。計算機の中に蓄えられたバーチャルな世界を CG 画像により提示する場合には、CG 画像を生成する際の視点位置や透視変換などの様々なパラメータを適切に設定することが重要となってくる。このようなパラメータはテレビ放送や映画などにおいて単眼用の、かつ実写映像との重ね合わせを必要としない純粋な CG 映像を生成する際にはあまり問題とされないが、両眼立体視を行う VR においては上記の両眼手がかりに直接影響することになるので、極めて重要かつクリティカルである。

立体感を誇張しない自然な三次元空間の提示を実現するためには、実環境へのトレイグジスタンスにおいては操作者側 HMD と同一の視野角・眼間距離を持つ撮像系により画像が生成され、カメラを搭載した頭部や作業を行う上肢などが操作者と完全に同一の運動を行うことが理想的である [60][31]。CG 画像を提示する場合においても実カメラが CG 画像生成の透視変換に置き換わるだけで理想条件は全く同一であり、その他の様々な条件を考慮して正確な立体画像を提示する方法が体系的に研究されている [50]。ところが、こうした正確な立体画像提示のための条件が完全に満足されない状況は、

実は簡単に成立しうる。撮像系・提示系のキャリブレーションが不十分な場合はもちろんのこと、例えば、アールキューブ構想 [16] における利用形態では、配置や光学系の異なる撮像システムを搭載した複数のロボットを単一のコンソールから切り替えて制御することが想定される。また、操作者と体格や構造の異なるロボットを制御する場合には、運動に関する同一性が確保できるとは限らない。また、周囲の広い領域を見回すことが必要とされるような種類のタスクにおいては、あえて同時に広角の領域が視野内に入ってきた方が望ましい場合も考えられる。このとき、対象とする視野角を物理的にカバーする HMD を利用することが理想的であるが、HMD はその名の通り液晶パネルや CRT などの提示デバイスを頭部に搭載する必要がある、現状では HMD に利用可能なデバイスの解像度が限られている。同じデバイスを用いて単純に光学系を広視野にしたのでは単位角度あたりの解像度、すなわち換算視力が低下するので、操作者はぼやけた不鮮明な画像を提示されることになって負担が大きい。

HMD を用いた立体画像提示に関しては、視野角と眼間距離の他にも、画像提示スクリーン像までの距離が固定であることに起因する焦点調節と輻輳の矛盾 [12]、光学系の歪み [49]、設計パラメータに現れない製作時の誤差および装着位置のずれによる誤差 [40]、瞳孔間距離の個人差 [42] およびその眼球運動による変化 [69] など、正確な立体知覚を阻害する様々な要因についての分析と対策に関する研究が行われている [51]。しかしこれらはデバイス単体あるいは頭部を固定した状況における静的な解析と評価であって、HMD を利用した環境提示の特徴である操作者の頭部運動（特に回転）を伴う場合に生じる状況については定量的な解析が行われていない。その一方で、VR システムにおける動的な要素として、頭部運動計測や CG 画像生成による時間遅延 [35] が存在する場合の影響と対策が報告されており [15][8][9][86]、この問題に関しては特に強調現実感 (Augmented Reality) [2]、複合現実感 (Mixed Reality) [39] の分野で活発に行われている。しかし、これらの研究における興味の対象は時間遅れであり、静的な立体視に関係する要素と動的な要素は個別に取り扱われてきた。

これに対し、本章では静的な立体視に関係するパラメータの不整合が、時間遅れの存在しない状況下で頭部運動を行った場合にどのような影響を及ぼすかについて考察を行う。このとき、前述のような立体視に関する数々の要因を一度に考慮して解析を行おうとすると、膨大な数のパラメータに左右される複雑な系を扱わねばならず、見通しのよい結果を得ることは困難である。そこで本章では、解析に用いるパラメータをあえて最も主要な眼間距離と視野角の 2 種類のみに絞り、光学系による歪み、製作時・装着時の誤差、瞳孔間距離の個人差や変動、あるいは輻輳と焦点調節の不一致などの空間知覚手がかりによる要因は対象外とする。

以上の条件設定の上で、本章ではまずマスタ（操作者側：画像提示系）とスレーブ（ロボット側：撮像系）との間で眼間距離および視野角の設定が異なる状況を想定し、その場合に生じる静的な空間歪みを幾何学的に分析して、それぞれのパラメータ不整合が及ぼす影響を直感的に理解しやすい結果として整理する。次に、このような状況下で操作者が頭部運動を行った場合に生じる操作者の空間認識への動的な影響を解析する。この影響は頭部運動を行う VR やテレイグジスタンスに特有の現象であり、静的な立体画像提示を行う場合は表面に現れない問題である。さらに、このようにマスタ・スレーブ間でパラメータの整合性が確保されない状況、特に視野角の不一致が存在する場合において、同時に頭部回転角倍率を制御することにより、頭部回転運動に対する注視点付近の画像の流れを適切に制御して作業性の低下を緩和する方法を提案する。最後に、これらの解析結果をふまえて作業実験により検証を行う。

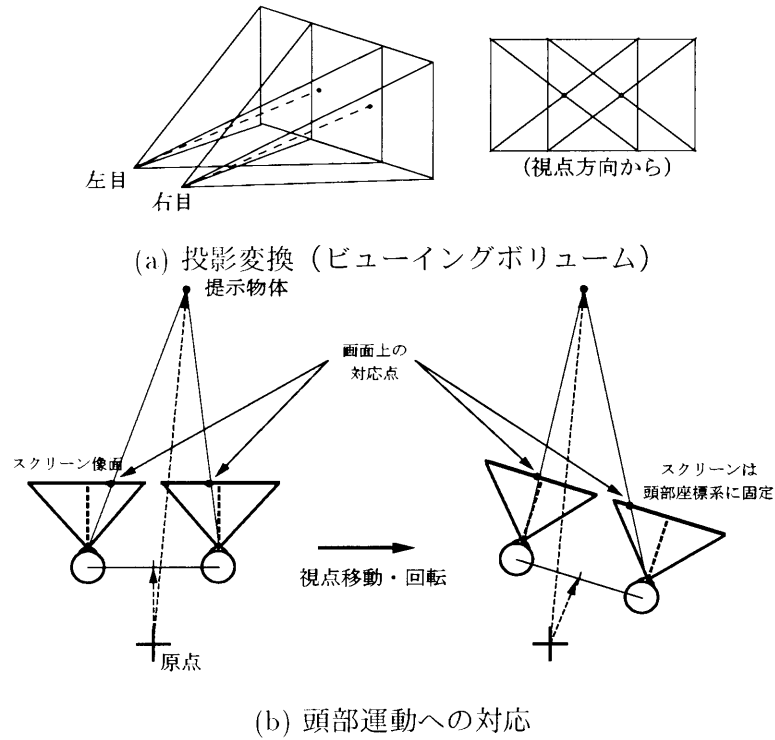


図 3.1 HMD を用いたシステムにおける投影変換と頭部運動

3.2 視覚・運動パラメータの整合性に関する幾何学的解析

3.2.1 HMD のパラメータ

HMD を使用する場合，操作者が運動しても目とスクリーンとの位置関係は変化せず，従って画像生成側の投影変換は常に一定である（図 3.1）．このような場合，バーチャル環境へのトレイグジスタンスを行う場合は常に一定の透視変換を用い，実環境へのトレイグジスタンスを行う場合は普通のカメラを用いて，それぞれバーチャル視点あるいはロボット頭部全体が操作者の運動に追従するよう制御すればよい．典型的な HMD では，スクリーン像が前額面に対して平行で，目からスクリーンに下した垂線の足がスクリーン中央となるように設計されている（無限遠点に対して 100%オーバーラップ）．この場合，光軸に対して上下左右対称な透視変換を用いるのが適当であり，設計上の主要パラメータは左右の視点間距離と視野角の 2 つである．

HMD におけるパラメータを Fig 3.2 に示す．両サイトの区別に関しては，リモート側を r ，オペレータ側を o で表し，左右に関しては右を R ，左を L と表す．リモート側，オペレータ側の片側視野角をそれぞれ φ_r, φ_o ，リモート側，オペレータ側の視点間距離をそれぞれ d_r, d_o と表記する．図 3.2 では，遠隔側の方が視点間距離が小さく，視野角が大きい状況を示している．操作者側に対する遠隔側の視野角倍率を β ，すなわち

$$\varphi_r = \beta \varphi_o \quad (3.1)$$

とし，また，片側視野角の正接の比を

$$\rho_\varphi = \frac{\tan \varphi_r}{\tan \varphi_o} = \frac{\tan \beta \varphi_o}{\tan \varphi_o} \quad (3.2)$$

と表しておく．それぞれのサイトにおいて左右視点の中間点を原点 O とし，前額面右方向を x 軸に，光軸方向を z 軸にとることとする．以下，簡単のため水平面内の振る舞いのみを扱うものとする．

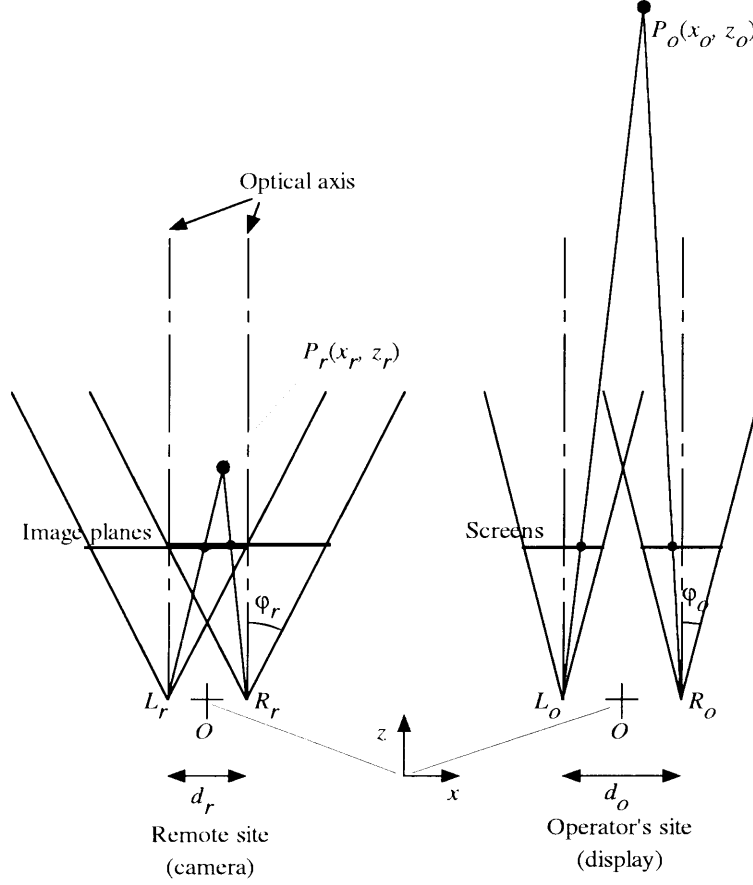


図 3.2 HMD を利用したシステムの視覚系パラメータ（視点間距離と視野角）

3.2.2 パラメータ不整合による空間の静的変形

ここで、遠隔/バーチャル環境側（画像生成系）と操作者側（画像提示系）で視点間距離あるいは視野角が異なる条件の下で、画像生成系で作成された画像がそのまま提示系のデバイスに提示される状況を考える．視点間距離が異なる場合はベースラインが異なるため、また視野角が違う場合には画像全体が光軸を中心に縮小・拡大されるため知覚される物体までの距離が実際と異なる．以下、この状況を幾何学的に解析する．

遠隔側において点 $P_r(x_r, z_r)$ を右、左の視点から見た角度をそれぞれ ψ_{Rr} , ψ_{Lr} と表すと、

$$\tan \psi_{Rr} = \frac{-x_r + d_r/2}{z_r} \quad (3.3)$$

$$\tan \psi_{Lr} = \frac{-x_r - d_r/2}{z_r} \quad (3.4)$$

と記述される．この点は遠隔側カメラにより左右それぞれの画像上の点として記録され、操作者側の提示デバイスにおいて視野中心からの距離が $1/\rho_\varphi$ 倍となる点として表示される．このため、操作者側において左右の視点から提示デバイス上の点を見た直線と光軸のなす角度は

$$\tan \psi_{Ro} = \frac{1}{\rho_\varphi} \tan \psi_{Rr} = \frac{1}{\rho_\varphi} \cdot \frac{-x_r + d_r/2}{z_r} \quad (3.5)$$

$$\tan \psi_{Lo} = \frac{1}{\rho_\varphi} \tan \psi_{Lr} = \frac{1}{\rho_\varphi} \cdot \frac{-x_r - d_r/2}{z_r} \quad (3.6)$$

なる角度 ψ_{Ro} , ψ_{Lo} となり, これらの直線の交点が操作者に知覚される点 $P_o(x_o, z_o)$ である. これらの直線の方程式は傾きと通る点の条件から

$$z = -\frac{x - d_o/2}{\tan \psi_{Ro}} \quad (3.7)$$

$$z = -\frac{x + d_o/2}{\tan \psi_{Lo}} \quad (3.8)$$

と表されるので, 連立させて解けば $P_o(x_o, z_o)$ が求められる. すなわち,

$$\tan \psi_{Lo} \left(x - \frac{d_o}{2} \right) = \tan \psi_{Ro} \left(x + \frac{d_o}{2} \right) \quad (3.9)$$

より,

$$(\tan \psi_{Ro} - \tan \psi_{Lo})x = -\frac{d_o}{2}(\tan \psi_{Ro} + \tan \psi_{Lo}) \quad (3.10)$$

しかるに, (3.6) 式より

$$\tan \psi_{Ro} - \tan \psi_{Lo} = \frac{d_r}{\rho_\varphi z_r} \quad (3.11)$$

$$\tan \psi_{Ro} + \tan \psi_{Lo} = -\frac{2x_r}{\rho_\varphi z_r} \quad (3.12)$$

であるから, P_o の x 座標は

$$x_o = \frac{d_o}{d_r} x_r \quad (3.13)$$

と求められる. オペレータサイトに対するリモートサイトの視点間距離の比を

$$\rho_d \equiv d_r/d_o \quad (3.14)$$

とおくと, x 座標は単に

$$x_o = x_r/\rho_d \quad (3.15)$$

となる. これは, x 軸方向の変形が視野角倍率に関係なく視点間距離倍率のみで決まることを示している.

これを直線の式 (3.8) に代入して z 座標 (奥行き) を求めると,

$$z_o = -\frac{x_o - d_o/2}{\tan \psi_{Ro}} = -\frac{x_r/\rho_d - d_r/2\rho_d}{\frac{1}{\rho_\varphi} \cdot \frac{-x_r + d_r/2}{z_r}} = \frac{\rho_\varphi}{\rho_d} z_r \quad (3.16)$$

と計算される. これは, z 軸方向に関しては視点間倍率に従って拡大・縮小されるとともに, さらに視野角倍率の従って拡大・縮小されることを示している.

以上をまとめると, リモート側の点 は, オペレータ側において点

$$P_o(x_o, z_o) = \left(\frac{x_r}{\rho_d}, \frac{\rho_\varphi z_r}{\rho_d} \right) = \frac{1}{\rho_d} (x_r, \rho_\varphi z_r) \quad (3.17)$$

にマッピングされる.

このように, 画像生成系と提示系で視点間距離および視野角が異なるシステムにおいては, 空間は単純な変形を伴って操作者に提示されることが示された. 視点間距離が異なる場合は空間全体が一様に拡大・縮小され, 視野角倍率が異なる場合にはさらに光軸方向に空間が拡大・縮小されることがわかる. この様子を図 3.3 に示す.

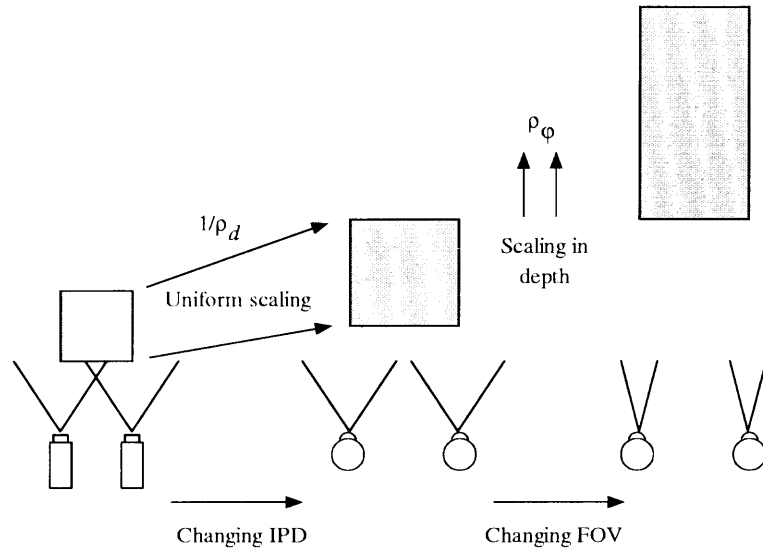


図 3.3 視点間距離と視野角の不整合による空間の静的変形

3.2.3 頭部回転運動による動的変形と物体の移動

前節の結果より，視点間距離のみが異なる場合には空間全体あるいは拡大・縮小あるいは逆に自分自身が縮小・拡大したものと見なして作業を遂行することができると考えられるが，視野角が異なる場合には頭部を回転させた場合に空間の圧縮・伸張の軸が頭部の方向と共に変化するので，空間が変形するように感じられる．この様子を Fig 3.4 に示す．上段左は広視野カメラを備えたりモットサイトの様子であり，これをオペレータ側で視野角の狭い HMD を用いて提示すると，下段左のように世界全体が遠方に伸張されて知覚される．

ここで，操作者が頭部を回転させ，それに追従して上段右のようにロボット頭部が回転した場合，操作者側では世界が下段右のように回転した頭部に対して前後方向へ伸張されるため，物体は細線で示した回転前の状態と比べて変形し，特別な点を除いて位置もずれる．このとき，操作者の正面にあった物体は変形しながら頭部の回転と同方向に移動するように感じられる．逆に，遠隔側が狭視野（望遠など）の場合には，物体は奥行き方向に縮小され，正面にあった物体は変形しながら頭部回転とは逆方向に移動するように感じられる．このような動的な変形は頭部回転運動を行った場合にはじめて観察されるものであり，従来頭部運動を伴わない両眼立体視システムにおいては重要視されなかった現象である．

この現象を幾何学的に分析する．図 3.4 において，簡単のため頭部は両眼の中央を中心に回転するものとする．実際に人間が頭部回転を行うときは回転中心が両眼の中央とはならないが，その場合は本節の解析結果に原点の平行移動による世界全体のシフト分を加味すればよい．さて，遠隔サイトにおける点 $P_r(x_r, z_r)$ が，操作者側では頭部回転前には点 $P_i(x_i, z_i)$ の位置に，回転後には点 $P_o(x_o, z_o)$ の位置に知覚されるものとする． P_r と P_i の関係は前節の結果より

$$P_i = \begin{bmatrix} 1 & 0 \\ 0 & \rho_\varphi \end{bmatrix} P_r \quad (3.18)$$

であり， P_r に対応する P_o の位置を求めるには，

1. 頭部回転中心に関して $-\theta$ 回転

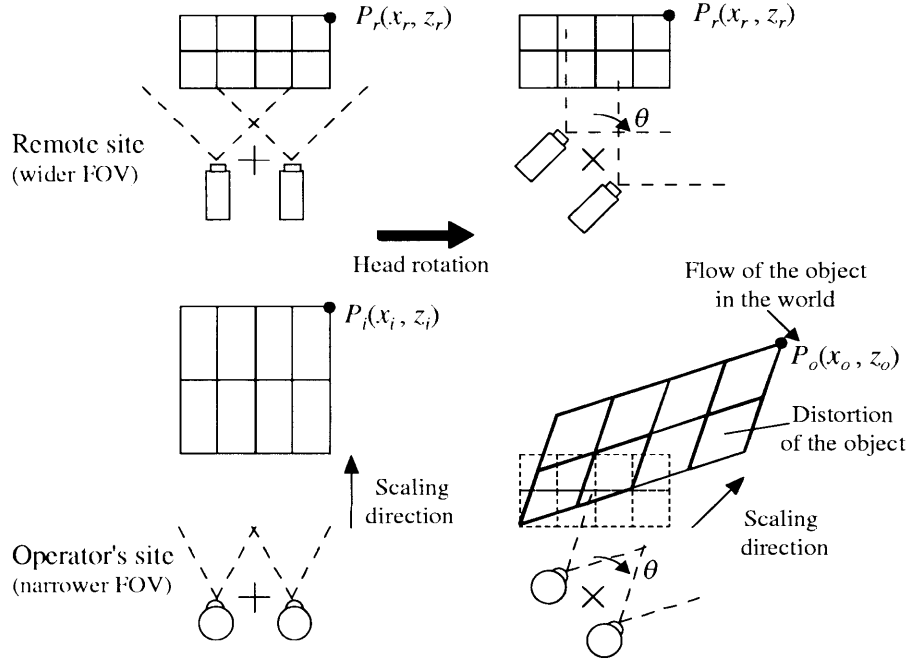


図 3.4 頭部運動による空間の動的変形

2. 奥行き方向に ρ_φ 伸張
3. 頭部回転中心に関して θ 回転を戻す

という操作を，原点に固定した座標系に関して行えばよい．つまり，

$$P_o = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \rho_\varphi \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} P_r \quad (3.19)$$

という関係が成り立つ．これを展開すると，

$$\begin{bmatrix} x_o \\ z_o \end{bmatrix} = \begin{bmatrix} x_r(\cos^2 \theta + \rho_\varphi \sin^2 \theta) + z_r(1 - \rho_\varphi) \sin \theta \cos \theta \\ x_r(1 - \rho_\varphi) \sin \theta \cos \theta + z_r(\sin^2 \theta + \rho_\varphi \cos^2 \theta) \end{bmatrix} \quad (3.20)$$

となる．微小回転角，すなわち $\theta \rightarrow 0$ のとき，

$$\begin{aligned} \begin{bmatrix} x_o \\ z_o \end{bmatrix} &\approx \begin{bmatrix} x_r + z_r(1 - \rho_\varphi)\theta \\ x_r(1 - \rho_\varphi)\theta + \rho_\varphi z_r \end{bmatrix} \\ &= \begin{bmatrix} x_r \\ \rho_\varphi z_r \end{bmatrix} + (1 - \rho_\varphi)\theta \begin{bmatrix} z_r \\ x_r \end{bmatrix} \\ &= \begin{bmatrix} x_i \\ z_i \end{bmatrix} + (1 - \rho_\varphi)\theta \begin{bmatrix} z_i/\rho_\varphi \\ x_i \end{bmatrix} \end{aligned} \quad (3.21)$$

と計算される．右辺第1項は回転前に操作者が知覚している点 P_i そのもので，第2項が点の移動に関する成分であり，これより各点の速度ベクトル

$$\vec{OP}_o = (1 - \rho_\varphi) \begin{bmatrix} z_o/\rho_\varphi \\ x_o \end{bmatrix} \frac{d\theta}{dt} \quad (3.22)$$

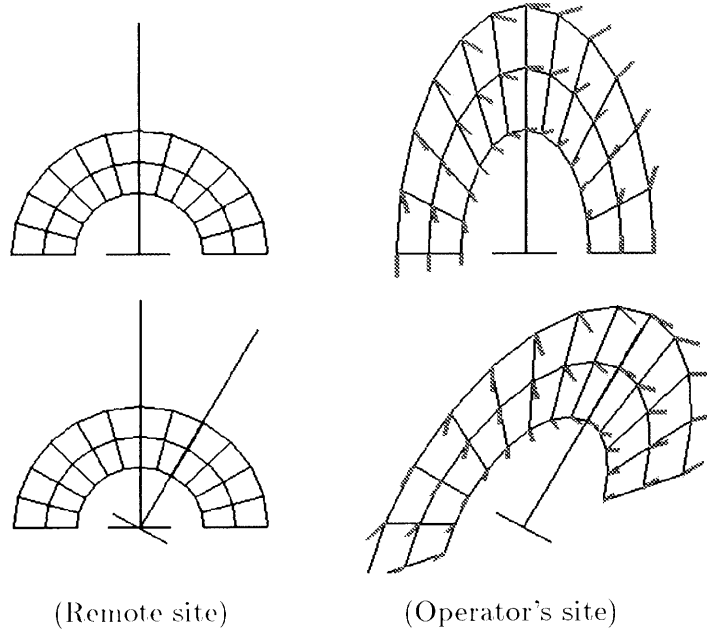


図 3.5 視野角の不整合による空間の動的変形に関するシミュレーション結果.

が導かれる．このベクトルの大きさは，カメラと HMD の視野角が異なる ($\rho_\varphi \neq 0$) の場合一般に零にならず，操作者の知覚上で点の移動が生じることが示される．この挙動を調べるため距離方向成分を計算すると，

$$\dot{\vec{OP}}_o \cdot \frac{\vec{OP}_o}{|\vec{OP}_o|} = \frac{1 - \rho_\varphi^2}{\rho_\varphi} \cdot \frac{x_o z_o}{\sqrt{x_o^2 + z_o^2}} \cdot \frac{d\theta}{dt} \quad (3.23)$$

である．この結果は，正中面方向の点 ($x_o = 0$) は奥行き方向に変化しないことを示している．また，提示画像の横方向の流れとして現れる回転方向成分は $(-z_o, x_o)$ 方向の単位ベクトルとの内積をとることにより，

$$\frac{1 - \rho_\varphi}{\rho_\varphi} \cdot \frac{\rho_\varphi x_o^2 - z_o^2}{\sqrt{x_o^2 + z_o^2}} \cdot \frac{d\theta}{dt} \quad (3.24)$$

と計算される．正中面付近，すなわち $|x_o| \ll 1$ において (3.24) 式の符号は $\rho_\varphi > 1$ のとき $d\theta/dt$ と同じであり，これは遠隔側視野角の方が大きい場合に正面付近の物体が頭部と同じ方向に流れることを意味する．逆に， $\rho_\varphi < 1$ のとき，(3.24) 式の符号は $d\theta/dt$ と逆になり，頭部とは反対の方向に流れることを示している．

頭部運動により知覚される世界が動的に変形し，点の移動が生じる様子を 図 3.5 に示す．図の左側は遠隔サイトにおけるロボット頭部の回転，右側は操作者側で知覚される世界の様子である．上段は正面を向いている瞬間（頭部を右方向へ回転中），下段は頭部が右へ回転した状態（やはり右方向へ回転中）を示す．各格子点に付加されている短い線分は，各点がその瞬間に移動する速度ベクトルの方向と相対的な大きさを示している．リモート側の視野角は操作者側視野角の 2 倍に設定されており，広角画像が HMD に縮小されて提示されている状況である．この状況では，世界が操作者に対して奥行き方向に伸長され，正面方向の物体は頭部運動と同じ方向にくっついて動くことが示される．

3.3 頭部回転運動の倍率制御による画像の安定化

3.3.1 安定化の原理

理想的な HMD システムの場合、すなわちロボットに搭載されたカメラあるいは使用する透視変換の視野角が HMD と等しく、(バーチャル) カメラの回転が操作者の頭部に対して正確に時間遅れなく制御されるならば、表示される画像の流れはちょうど操作者の頭部運動を打ち消すものとなり、操作者がどのように頭部を回転させるかに関係なく世界全体は静止して見える。これは人間の前庭動眼反射 (VOR: vestibulo-ocular reflex) [44] のメカニズムと矛盾しない視覚提示が行われていることを意味する。しかしながら、これまでの議論で、撮像系と提示系で視野角が異なる場合には、空間の変形により提示物体の空間的な位置が不自然に移動することが示された。この現象は頭部運動による前庭感覚と提示される視覚との不一致を生み、simulator sickness の直接的な引き金となる。

画面全域にわたり提示物体の安定化を行うことは困難であるが、視野角が異なる場合にリモート側の頭部回転運動倍率を制御することにより、特定の注視点近傍における画像の流れの振る舞いを正常な場合と同様のものにすることは可能である。以下、このための条件を 2 通りの観点から導出し同一の結果が得られることを示す。

これらの方法の基本的な原理は、遠隔ロボットあるいはバーチャル人間の頭部回転運動をある一定の倍率で拡大・縮小することにより、提示される画像の流れが通常の（操作者側と遠隔側の視野角が等しく、頭部運動が一致するように制御される）状態と等価になるようにすることである。この倍率は、操作者の前庭動眼反射の特性に適合するように設定する。視野角が異なる限り提示される世界全体が歪んで見えることには変わりないが、この条件が満たされれば、操作者はある点を見つめる限りその点は動かず静止しているように感じられる。

3.3.2 スクリーン上の点の流れに基づく導出

3.2.3 節で述べた現象を、頭部回転運動を行った場合における画像上の点の流れに注目して解析してみる。図 3.6 において、(a) はカメラと HMD の視野角が一致しておりリモート側のロボット頭部回転角が操作者のそれと等しくなるよう制御した場合、すなわち理想的な条件である。ここで、操作者が頭部を右へ回転させると提示される画像の中の点は左へ移動する。撮像系と提示系の視野角が等しい条件のままで (b) のようにロボットの頭部回転角が操作者の頭部回転角の α 倍になるように制御すると、 $\alpha > 1$ の場合は画像中の点の移動量は通常の場合より大きくなる。(c) のように遠隔側の視野角が HMD の β 倍の時は、 $\beta > 1$ ならばカメラによって得られる画像は縮小されて HMD に表示され、その結果頭部運動に伴う画像の流れは通常の場合より小さくなる。(d) では頭部運動角と視野角が両方スケールアップされている。この場合、頭部運動角倍率 α による画像の流れの拡大・縮小と視野角倍率 β による画像の流れの縮小・拡大がちょうどキャンセルするように倍率を決定すれば、頭部運動に起因する画像の流れは通常の場合と等しくなり、一点注視時における VOR 成立の条件が満たされることになる。

さて、このように視野角倍率が β の時に静止している物体が静止して見えるための頭部回転角倍率 α の値を求める。

以下、次の記号を用いる。

D : 視点とスクリーンの距離

x : スクリーン中心から注視点までの距離

ψ : 原点と注視点を結ぶ線が光軸方向となす角

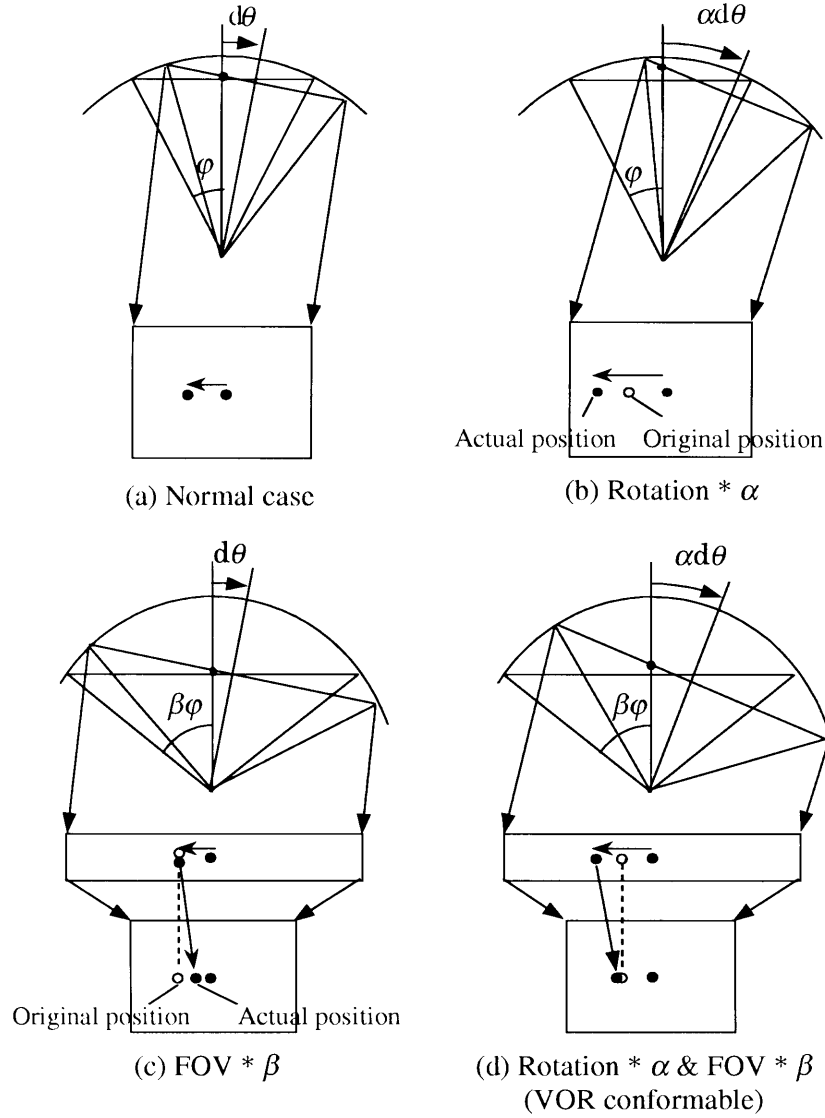


図 3.6 視野角および回転角の相違による画像上の点の流れ

η : ψ の正接 ($\eta = \tan \psi = -x/D$)

また、通常の（理想的な）ケースを i , 視野角・回転スケーリングを行う場合の遠隔サイトを r , 操作者サイトを o という添字を使って表現するものとする。

図 3.7 左（通常の場合）において、操作者は光軸方向となす角が ψ_i であるスクリーン上の点 x_i を注視しているものとする。頭部が視点を中心に微小角 $d\theta$ 回転すると、スクリーン上の点の位置は

$$dx_i = R_i d\theta / \cos \psi_i \quad (3.25)$$

だけ移動する。ここで、 R_i は視点（原点）と点 x_i との距離であり、

$$R_i = \sqrt{D^2 + x_i^2} = D / \cos \psi_i \quad (3.26)$$

であるから、

$$dx_i = \frac{D d\theta}{\cos^2 \psi_i} \quad (3.27)$$

ここで、視野角・回転角スケーリングを行う場合に操作者が視野角の等しいケース（左）と同じ点を注視している，すなわち $x_o = x_i$ （したがって $\psi_o = \psi_i, \eta_o = \eta_i$ ）とすると，

$$\cos^2 \psi_i = \frac{D^2}{D^2 + x_i^2} = \frac{1}{1 + (x/D)^2} = \frac{1}{1 + \eta_o^2} \quad (3.34)$$

$$\cos^2 \psi_r = \frac{D^2}{D^2 + x_r^2} = \frac{1}{1 + (x_r/D)^2} = \frac{1}{1 + \rho_\varphi^2 \eta_o^2} \quad (3.35)$$

と表される．よって回転角倍率は

$$\alpha = \rho_\varphi \frac{1 + \eta_o^2}{1 + \rho_\varphi^2 \eta_o^2} \quad (3.36)$$

と導かれる．視野中心の場合， $\eta_o = 0$ であるから，単純に

$$\alpha = \rho_\varphi \quad (3.37)$$

となる．

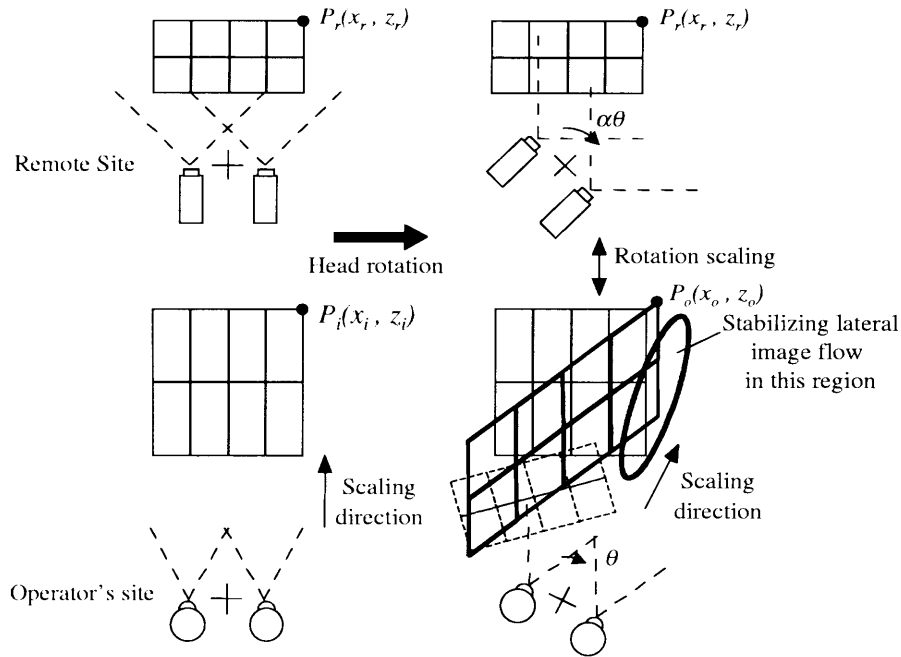


図 3.8 頭部回転運動の倍率制御による画像安定化の原理

3.3.3 両眼立体視時の空間変形に基づく導出

空間変形の解析に基づいた注視点近傍の画像安定化の原理を図 3.8 に示す．この図では，遠隔側の視野角が操作者側の視野角よりも大きい場合（視野角倍率 $\beta > 1$ ）を示している．遠隔側と操作者側の視野角が異なり，世界が横方向に圧縮されて表示され，光軸方向に伸長されて知覚されることに関しては図 3.4 と同じであるが，操作者の頭部回転運動と比べて遠隔側の頭部を大きく回転させているところが異なる．

以下，注目している点の方向が一定となるための条件を考える．操作者の頭部が θ 回転し，遠隔側のロボット頭部が $\alpha\theta$ 回転するものとする．すると，3.2.3 節の (3.19) 式における最初の $-\theta$ 回転

操作（右側の行列）が $-\alpha\theta$ 回転に置き換えられるので，回転後の点の位置は

$$P_o = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \rho_\varphi \end{bmatrix} \begin{bmatrix} \cos \alpha\theta & \sin \alpha\theta \\ -\sin \alpha\theta & \cos \alpha\theta \end{bmatrix} P_r \quad (3.38)$$

により計算される．3.2.3 節と同様にこれを展開すると，

$$\begin{bmatrix} x_o \\ z_o \end{bmatrix} = \begin{bmatrix} x_r(\cos \theta \cos \alpha\theta + \rho_\varphi \sin \theta \sin \alpha\theta) \\ + z_r(\cos \theta \sin \alpha\theta - \rho_\varphi \sin \theta \cos \alpha\theta) \\ x_r(\sin \theta \cos \alpha\theta - \rho_\varphi \cos \theta \sin \alpha\theta) \\ + z_r(\sin \theta \sin \alpha\theta + \rho_\varphi \cos \theta \cos \alpha\theta) \end{bmatrix} \quad (3.39)$$

となる．微小回転角，すなわち $\theta \rightarrow 0$ のとき，

$$\begin{aligned} \begin{bmatrix} x_o \\ z_o \end{bmatrix} &\approx \begin{bmatrix} x_r + z_r(\alpha - \rho_\varphi)\theta \\ x_r(1 - \alpha\rho_\varphi)\theta + \rho_\varphi z_r \end{bmatrix} \\ &= \begin{bmatrix} x_r \\ \rho_\varphi z_r \end{bmatrix} + \theta \begin{bmatrix} (\alpha - \rho_\varphi)z_r \\ (1 - \alpha\rho_\varphi)x_r \end{bmatrix} \\ &= \begin{bmatrix} x_i \\ z_i \end{bmatrix} + \theta \begin{bmatrix} (\alpha/\rho_\varphi - 1)z_i \\ (1 - \alpha\rho_\varphi)x_i \end{bmatrix} \end{aligned} \quad (3.40)$$

と計算され，速度ベクトル

$$\dot{\vec{OP}}_o = \begin{bmatrix} (\alpha/\rho_\varphi - 1)z_o \\ (1 - \alpha\rho_\varphi)x_o \end{bmatrix} \frac{d\theta}{dt} \quad (3.41)$$

の形に整理される．

頭部回転による変形によって空間中の点の方向が変化しないためには，速度ベクトルの方向が直線 OP_o と一致しなければならないので，

$$\frac{(\alpha/\rho_\varphi - 1)z_o}{(1 - \alpha\rho_\varphi)x_o} = \frac{x_o}{z_o} \equiv -\tan \psi_o \quad (3.42)$$

であり，3.3.2 節と同様に

$$\eta_o = \tan \psi_o \quad (3.43)$$

とおくと，

$$\frac{\alpha/\rho_\varphi - 1}{1 - \alpha\rho_\varphi} = \eta_o^2 \quad (3.44)$$

より

$$\alpha = \rho_\varphi \frac{1 + \eta_o^2}{1 + \rho_\varphi^2 \eta_o^2} \quad (3.45)$$

という結果が導かれる．これは，スクリーン上の点の流れに基づく解析結果 (3.36) 式と一致する．

また，常に正中面方向の点を横方向に静止させるように頭部回転運動の倍率を制御する場合 ($\alpha = \rho_\varphi$)，空間中の点の速度ベクトルは (3.41) 式より，

$$\dot{\vec{OP}}_o = \begin{bmatrix} 0 \\ (1 - \rho_\varphi^2)x_o \end{bmatrix} \frac{d\theta}{dt} \quad (3.46)$$

となるので，すべての点がそれぞれの瞬時に於いて奥行き方向のみに移動することがわかる．特に，正中面上の点においては $x_o = 0$ であるから，速度の z 成分も 0 となり，左右，奥行き方向ともに静止しているように観察されることが示される．これは，常時正中面方向付近を注視するタスクの場合，遠隔ロボットの頭部回転運動を視野角倍率に従って拡大/縮小して制御すればよいことを示唆している．この様子を図 3.9 に示す．頭部が回転した場合，世界全体は初期状態と比較して奥行き方向に歪むが，正中面付近の点のフローが抑えられている様子が確認される．

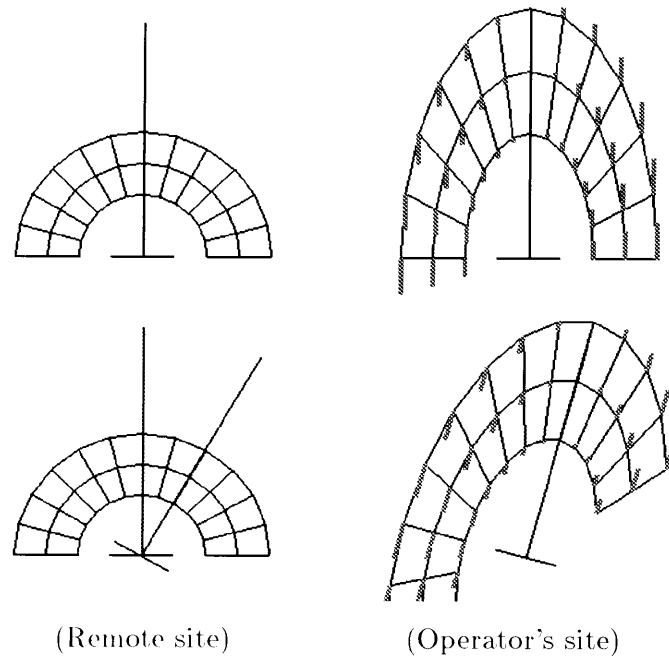


図 3.9 回転角倍率のスケーリングによる画像の安定化（シミュレーション結果）

3.4 評価実験

3.2 節および 3.3 節で示した，画像生成系と提示系のパラメータ不整合による影響と安定化の効果を検証するため，作業実験を行った．前述した幾何学的解析において，視覚パラメータのうち眼間距離は並進運動に関係し，視野角は並進運動と回転運動の両方に影響する．これらの視覚パラメータと運動パラメータの組み合わせにより，運動感覚と提示される視覚刺激との間で整合性の確保される状況，あるいはされない状況を設定する．

3.4.1 実験装置

このようなパラメータ変化による影響を実験的に検証する場合，バーチャル環境を用いるとパラメータの制御が容易であるため，ここではこのアプローチを用いる．しかしながら，本研究の目的でバーチャル環境を利用するにあたっては，操作者の運動を実時間で計測してバーチャル環境の更新を行い即座に視覚提示へ反映させることにより，時間遅れの影響を排除することが要求される．このため，第 2 章で記述した実験システムを用いて，システム全体の時間遅れの最小化を図った．

3.4.2 実験 1: 並進パラメータに関する評価

まず，並進方向のに関するパラメータを制御し，作業実験を行った．制御する量は眼間距離と，操作者の体幹運動に起因する頭部並進運動の倍率および手先位置の並進運動倍率である．作業者はトレイグジスタンスマスタ装置の座席に着席し，頭部および手先を自由に動かせるようになっている．本実験では，体幹運動に起因する頭部並進運動の倍率と，手先位置の運動倍率をコントロールし，作業性の評価を行う．

- (a) 通常の場合，すなわちバーチャル環境側の頭部運動および手先位置が操作者のそれと等しくなる場合．

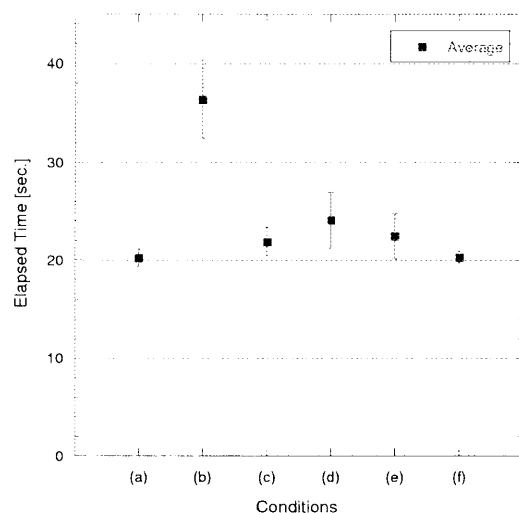


図 3.10 実験 1: 作業所要時間の平均値と標準偏差

- (b) バーチャル環境側の頭部並進運動量を実際の 4 倍にした場合.
- (c) バーチャル環境側の頭部並進運動量をゼロにした場合.
- (d) バーチャル環境側の眼間距離を実際 (64[mm]) の 2 倍にした場合.
- (e) バーチャル環境側の手先並進運動量 (腕の長さ) を実際の 1.5 倍にした場合.
- (f) 頭部並進運動, 眼間距離, および腕の長さをそれぞれ 1.5 倍にした場合.

これらの条件のうち, 最後の (f) は眼間距離, 頭部並進運動, 手先位置のすべてが同じ倍率で拡大されており, 周囲の環境と比較して自分自身が大きくなったかのような感覚を有して作業することが可能である. しかしながら, (a) と (f) を除く他のすべての条件においては提示される視覚と運動感覚との間に何らかの不整合が存在する.

評価は決められた作業タスクを遂行するために要した時間を計測することにより行った. タスクは, 初期位置から左手ジョイスティックを用いて作業台のある位置まで移動し, 台の上の 3 個の積み木を右手により積み上げ, そのあと積み木が 1 個ずつ台の上に置かれている状態に戻すというものである.

実験は, それぞれの場合について作業完了までの所要時間の計測を 5 回ずつ行ない, 平均値をとって比較した.

ある被験者 (A) が行なった, それぞれの条件における作業所要時間の平均値と標準偏差を図 3.10 に示す. 平均値を ■ 印, これを中心に標準偏差の範囲を I 印で示している. バーチャル環境側と実操作者のパラメータを一致させた場合 (a) と他の場合 (b) ~ (f) とで作業時間の平均値を比較すると, (f) を除いて作業性が悪化していることがわかる.

バーチャル人間と実オペレータのパラメータを一致させた場合 (a) と他の場合 (b) ~ (f) とで作業時間の平均値を比較すると, (f) を除いて作業性が悪化していることがわかる.

作業時間の差を統計的に評価するため, それぞれの条件における所要時間差が有意であるかどうかを t 検定にかけた. その結果を表 1 に示す.

(a) と他の場合との組合せでは, 危険率 5% ($t = 2.776$) で (b), (c), (d) が有意であり, (e), (f) が有意でないと検定された.

表 3.1 実験 1: 所要時間差に関する t-検定の結果

	(b)	(c)	(d)	(e)	(f)
(a)	9.7649	3.4428	3.2950	2.0579	0.1804
(b)	—	9.1126	8.0768	6.3175	7.5183
(c)	—	—	3.0832	0.7912	2.2967
(d)	—	—	—	1.9025	2.7927
(e)	—	—	—	—	2.3325

別の被験者 (B) が作業を行なった結果は、作業時間の平均値は被験者 (A) と同様の傾向が見られるものの、作業時間のばらつきが大きく、t-検定の結果は危険率 5% でいずれの場合も有意とは認められなかった。

また、被験者が作業を行なった際の感想として、(b) は頭の位置が定まらず非常に作業しづらい、(c) はあまり違和感を感じない、(d) は腕や対象物が不自然に小さく感じられる、(e) は腕が長く感じる、(f) は比較的自然的な感じがするが対象物が小さく感じられる、といった内観報告が得られた。

被験者 (A) の結果について、(a) とその他の条件とを比較する。(b) は、特に目立った差が観察され、操作感でも作業のしづらさが報告された。これはパラメータの変化量が 4 倍と大きいことも関係しているものと思われるが、上体運動に伴う頭部並進運動量が実際のそれと対応していないと、臨場感を損なうことがわかる。(c) は、頭部が自然に並進運動することにより作業が容易になることを示している。(d) は、「顔が大きい」場合に相当するが、人間とサイズの異なるものへテレレイグジストする場合に、単に視覚器の位置を変更しただけでは作業しづらくなることを示している。(e) は、大きさの異なるマニピュレータを持つロボットなどを利用したテレレイグジスタンス作業を行う場合、注意が必要であることを示している。(f) は、(a) と作業時間の相違がほとんど見られない。(b)、(c)、(d) と比較すると、いずれも作業時間差が有意である。これは、マイクロロボットを利用したテレレイグジスタンスなど、スケールの異なる世界へテレレイグジストする場合に、(b)、(d)、(e) のように部分的なスケールリングを行なうのではなく、遠隔/バーチャル環境側と操作者側が完全に相似となるようパラメータを設定することにより、自分がその大きさの存在になったかのような、自然な臨場感を得ることができる可能性を示している。

一方、本実験では被験者間の作業時間のばらつきが大きく、被験者によっては各条件間の有意性を検出することができないという結果も得られた。これらは、実験で行なった積木作業自体が、一旦作業可能範囲への移動を行なってしまえばそれほど頭部の運動を必要としないことに起因するのではないかと考えられる。

3.4.3 実験 2: 回転運動に関する評価

頭部回転運動に関するパラメータの整合性を評価するため、ターゲット追従実験を行った。

バーチャル空間において十字カーソルを被験者の頭部正面 1m に固定し、被験者は頭部を回転させて移動するターゲットにカーソルを追従させる。ターゲットは被験者の頭部を中心とする半径 1[m] の球の表面上を移動する。ターゲットの運動は、ランダム信号をローパスフィルタに通した波形により制御される。ターゲットおよび被験者の頭部運動回転角を記録し、角度波形の比較を行った。実験では、次の 4 通りの場合について比較を行った。

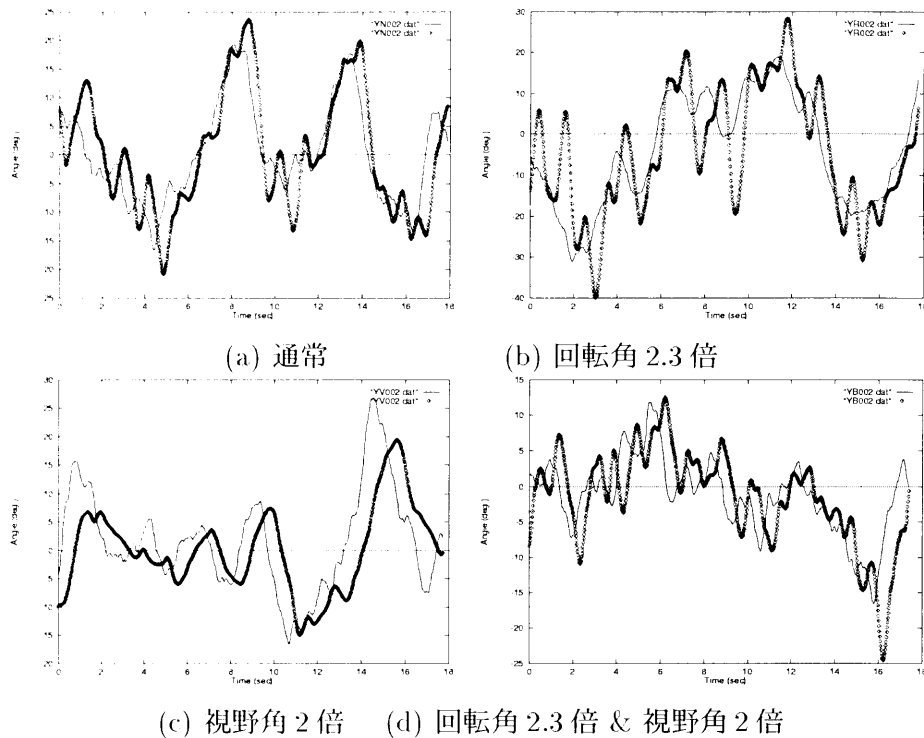


図 3.11 実験 2: 頭部回転追従運動の波形

- (a) 通常の場合：バーチャルカメラの視野角を HMD の視野角と等しく設定し ($\beta = 1$)，バーチャルロボット頭部回転運動角が被験者の頭部回転運動角と等しくなるよう制御する ($\alpha = 1$)．
- (b) 回転運動のみをスケールリング：バーチャルカメラの視野角は HMD と等しく設定し ($\beta = 1$)，頭部回転運動角倍率のみを 2.3 倍とする ($\alpha = 2.3$)．この 2.3 という値は，使用している HMD の視野角が 40 度で，バーチャルカメラの視野角が 80 度である場合 ($\beta = 2$) を想定して 3.36 式により計算されたものである．
- (c) 視野角のみをスケールリング：バーチャルカメラの視野角を HMD の 2 倍に設定し ($\beta = 2$)，頭部回転運動角は被験者の頭部回転運動角と等しいままにする ($\alpha = 1$)．
- (d) VOR に適合する場合：視野角と頭部回転運動角をともにスケールリングする．すなわち，バーチャルカメラの視野角を 80 度とし ($\beta = 2$)，頭部回転運動角倍率を 2.3 倍に設定する ($\alpha = 2.3$)．

図 3.11 にそれぞれの場合のターゲットと被験者の頭部運動の運動波形 (yaw 角) を示す．各々のグラフにおいて，実線はターゲット，印付きの線は頭部の運動を示す．通常の場合と比較して，回転運動のみをスケールリングした場合 (b) はオーバーシュートが大きくなっている．これは被験者がターゲットに追従するため頭部を回転させるとバーチャルロボットの頭部が必要以上に大きく回転し，その結果行き過ぎてしまうと解釈される．視野角のみをスケールリングした場合 (c) ではアンダーシュートあるいは時間遅れが大きくなっていることが確認される．これは，被験者がターゲットの移動（提示画像内では縮小されて見える）に対して普通に頭部を回転させても必要な回転量が得られず，さらに余分に回転させなければならない状況が生じていると解釈される．これらに対し，視野角および回転運動の両方をスケールリングし VOR に適合する条件 (d) では，通常の場合と比較して被験者の挙動に大きな差異は認められない．

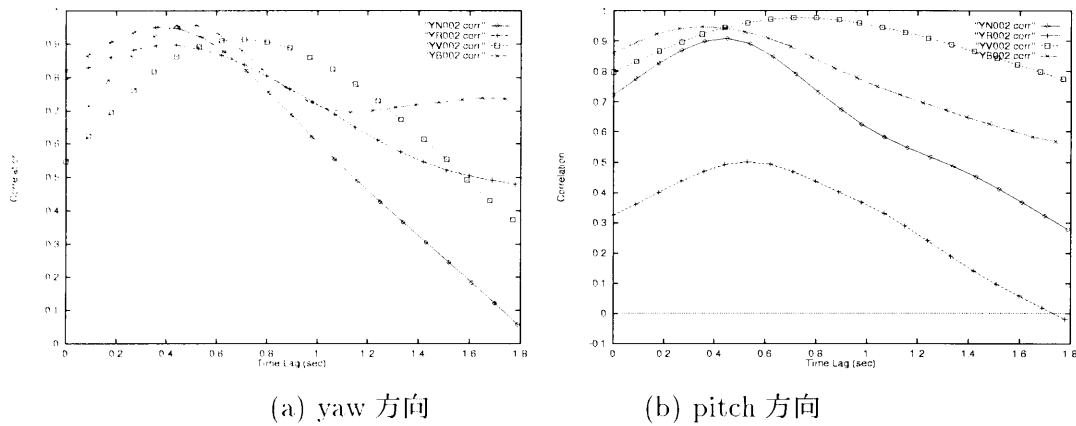


図 3.12 実験 2: 頭部回転追従運動の相互相関

各々の条件での被験者の挙動を定量的に評価するため、ターゲットと被験者頭部の回転運動角の相互相関関数を計算した。ヨー角、ピッチ角の相互相関関数を Fig 3.12 に示す。各々のグラフで、◇印は通常の場合、+印は回転角のみをスケーリングした場合、□印は視野角のみをスケーリングした場合、×印は視野角と回転角の両方をスケーリングした場合（VOR 適合条件）である。相互相関関数の特徴量として、以下の 2 種類の値について比較を行う。

- 極大値：追従タスクの正確さ
- 極大値をとる時間遅れ：被験者の応答の時間遅れ

これらの値を表 3.2 に示す。通常の場合と比較して、回転角もしくは視野角のどちらか一方のみをスケーリングした場合はそれぞれ極大値、時間遅れに被験者応答特性の悪化が見られるが、両方を VOR 成立条件に適合するようスケーリングした場合は極大値、時間遅れともに通常の場合と遜色ない結果が得られた。

表 3.2 実験 2: 頭部回転追従実験の相互相関極大値と時間遅れ

	Yaw		Pitch	
	Max.	Lag	Max.	Lag
(a)	0.95	0.36	0.91	0.45
(b)	0.90	0.36	0.50	0.53
(c)	0.91	0.71	0.98	0.80
(d)	0.96	0.52	0.95	0.35

3.5 まとめ

HMD を利用したトレイグジスタンスおよびバーチャルリアリティシステムにおいて、遠隔/バーチャル環境側と操作者側の眼間距離および視野角が異なる場合の空間認識への影響を幾何学的に分析し、眼間距離の不整合は世界全体の拡大・縮小として、視野角の不整合は奥行き方向の拡大・縮小と

して認識されることを明らかにした．この結果から，視野角の不整合が存在するシステムにおいて頭部運動を行った場合，世界が動的に歪んで観察されるとともに，不自然な提示画像のフローにより simulator sickness を起こしやすい状況が生じることを指摘した．さらに，この状況が避けられないような場合にもある程度の操作性を維持するため，頭部回転運動倍率を制御することにより人間の前庭動眼反射のメカニズムに適合する自然な画像の流れを実現する方法を考案した．以上の解析を検証するため，並進方向に関係するパラメータと回転方向に関係するパラメータの双方について作業性評価実験を行い，これらの整合性が成立することの重要性を示した．

画像提示デバイスを用いて正確な立体画像提示を実現することと，時間軸方向の動的な振る舞いは従来別々に取り扱われてきたが，本章で示したように，静的な立体画像提示の整合性が満たされていない場合は，たとえ時間遅れが皆無という理想的なシステムにおいても不自然な画像の流れにより不都合が生じることが明らかになった．これらの整合性を確保することはデバイス単体ではなくシステム全体を設計・構築する時の仕事であり，個々のデバイスのスペックとしては表現されないものである．しかし，このシステム全体としての特性は直接的に操作者へ影響を与えるものであり，作業への適用可能性や臨場感に対する一つの指標として考慮されるべきものである．

第 4 章

視覚提示デバイスの特性分析

前章までは、頭部搭載型ディスプレイ (HMD) を用いたシステムの構築とその挙動に関する議論を展開した。本章では、さらに高いレベルでの視覚と運動感覚の整合性を指すため、第 1 章で紹介した各視覚提示方式のうち、特にデバイス/スクリーンとユーザの視点との間の位置関係に関する分類に着目し、それぞれの方式が持つ性質に関して分析する。

4.1 視覚提示デバイスの種類

1990 年代初頭においては、トレイグジスタンスあるいは VR 用のデバイスといえば HMD が代表格であり、操作者がアクティブに頭部運動を行って周囲を見回すことにより遠隔/VR 空間を体験するというスタイルが定着していた。この見回し動作は、HMD の物理的視野角が狭いために同時には VR 空間のうちのほんの一部の情報しか得られないことをカバーするものであり、磁気センサや機械リンク式位置・姿勢計測装置などによる操作者の頭部運動計測とそれに実時間で対応した提示画像の更新は必須である。

一方で、三次元画像提示の分野では、据え置き型のディスプレイ装置を用い、特に頭部運動計測を行わずに立体空間の提示を行う事例が数多く存在する。これらは主に「立体テレビ」への応用を狙ったものであり、いわば操作者の存在する空間にあいた VR 空間への窓と言うべき形態である。頭部運動計測を行わない場合は、当然視点を反映した画像更新が行われないので、提示される空間の立体感は不正確で歪んだものになる。

こうした中、1992 年に米国イリノイ大学の DeFanti らが CAVE (CAVE Automatic Virtual Environment) [6] と呼ばれるシステムを開発・発表した。これは大型のスクリーンとプロジェクタの組合せを立方体の部屋状に構成し、操作者の周囲全体を提示画像で覆ってしまうというものである。各スクリーンに提示される映像は液晶シャッター眼鏡により立体提示を行うようになっており、完全に立体画像の中に没入することができる。CAVE は図 4.1 に示すように 4 面で構成され、正面、左右面は背面投射、下面は天井方向からの前面投射により提示している。CAVE はその後 FakeSpace Systems 社の商品として市販されるようになった。

また、1997 年には東京大学インテリジェント・モデリング・ラボラトリー (IML) 内に 5 面のスクリーンを持つ CABIN (Computer Aided Booth for Image Navigation) [18][18] [18][20] (図 4.2) が建設された。これは背面を除く正面、左右、上下にスクリーンを配置したもので、すべて背面投影で行う。CAVE と異なり床面の映像投影を天井から行うことができないため、床面に強化ガラスを用いて床ごと持ち上げ、床下からの背面投影を実現している。さらに 1998 年、岐阜県科学技術振興センター (テクノプラザ) に背面をも含む 6 面のシステム COSMOS [71] (図 4.3) が構築され、人間の周囲を映像で完全に包囲する環境が実現されている。COSMOS は CABIN の発展形であり、全面を背面投影で行い、高床式の構造を持つとともに、磁気センサの使用を考慮してスクリーン周囲のフ

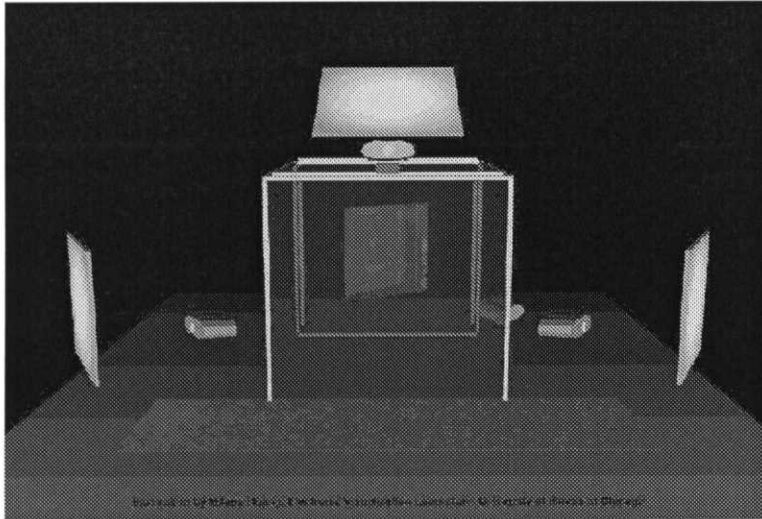


図 4.1 CAVE[6] の全景．(イリノイ大学 EVL の Web ページより引用)

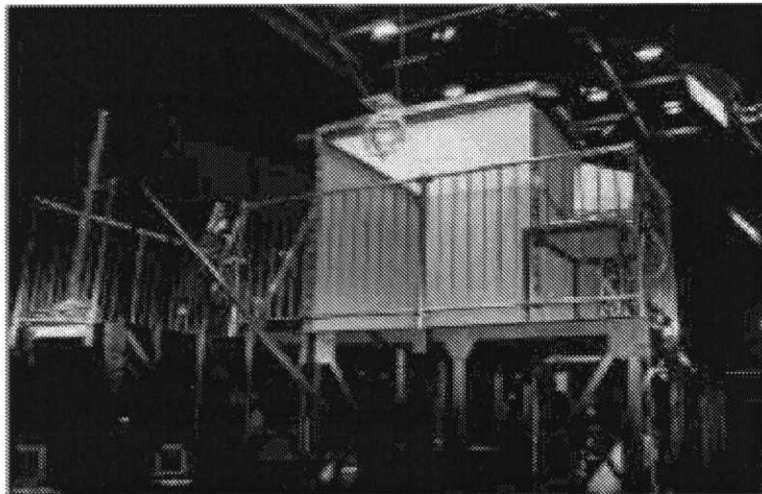


図 4.2 CABIN[18] の全景（東京大学 IML の Web ページより引用）．

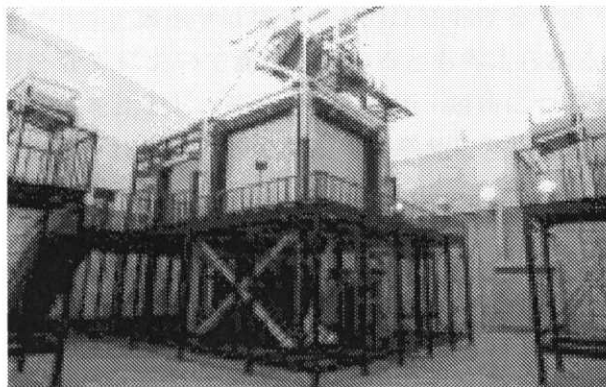


図 4.3 COSMOS[71] の全景（岐阜県科学技術振興センターの Web ページより引用）．

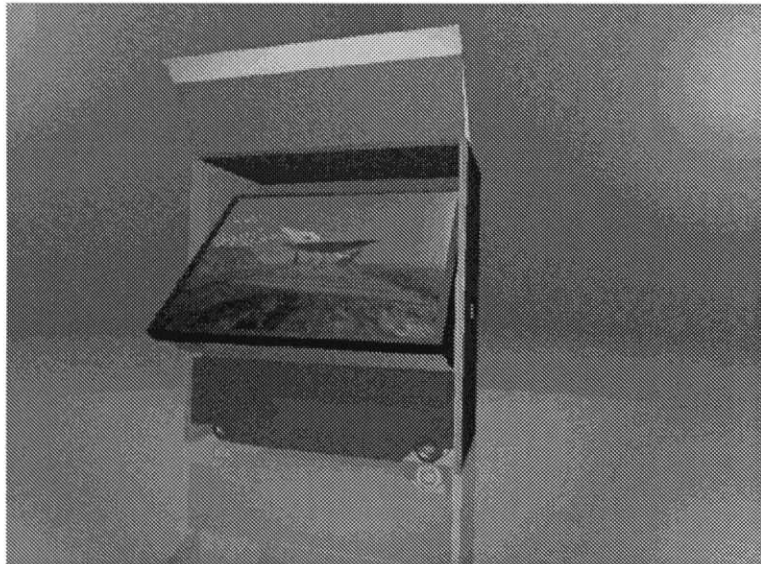


図 4.4 ImmersaDesk システム（イリノイ大学 EVL の Web ページより引用）。

レーンを木造にするなどの改良が行われている。

多面ばかりでなく、1面のシステムについても様々なシステムが開発されている。まず、CAVEの開発を行ったイリノイ大学において ImmersaDesk というシステムが開発され、これも FakeSpace Systems 社から市販されている。ImmersaDesk は基本的に CAVE の 1 面版であり、80 インチほどの大型スクリーンを斜めに配置しプロジェクタから背面投影により画像を提示している。また、ドイツにおいては会議テーブルの机面のような位置にスクリーンを配置し映像を提示する Responsive Work Bench [29] が開発されている。Responsive Work Bench は 2 人の頭部位置に連動したステレオ画像を提示できることが特徴的である。

これらの大型スクリーンとプロジェクタを利用したシステムは総称して Immersive Projection Technology (IPT) と呼ばれる。また、スクリーンは据え置きであるが頭部運動を提示画像に反映させるという意味では Head-Trackted Display (HTD) とも呼ばれる。IPT, HTD は近年各地で建設・設置が進められており、その視覚提示効果は極めて良好であるという評価が定着しつつある。特に多面 IPT は非常に大規模なシステムであり、構築には巨大な空間とコストを要するが、それにもかかわらず建設事例が増えていることは、サイズとコストのデメリットを超えるメリットが存在することを示唆している。IPT におけるメリットは、裏返せば従来の HMD におけるデメリットであり、HMD を利用したシステムにおける不満を IPT において満たしているとも考えられる。

以下、本章では HMD と IPT を比較し、それぞれの長所・短所について分析を行う。特に、操作者が頭部運動を行った場合に提示されるべき画像の変化の挙動に注目する。

4.2 HMD と IPT の長所と短所

これまで一般に、HMD に対する IPT の利点として以下のような点が言及されてきた。

1. 広視野角：特に CAVE などの多面 IPT では、操作者の視野全域を映像で覆うため、完全に VR 空間の中に没入することができる。
2. 高解像度：ワークステーション級の解像度の画像を表示するため、高精細な VR 空間の表示が

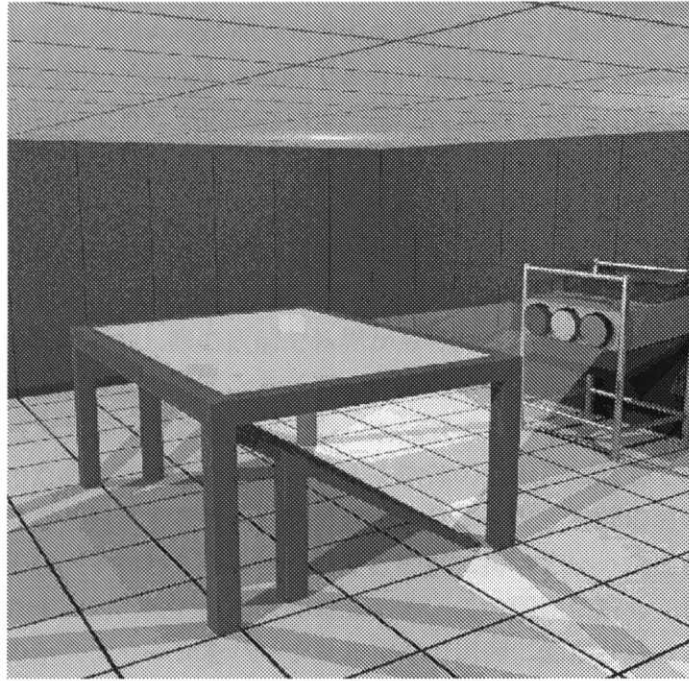


図 4.5 ResponsiveWorkBench システム（スタンフォード大学の Web ページより引用）。

可能。

ここでは、これらの点に関して再検証を試みる。

まず、広視野角に関してであるが、CAVE, CABIN, COSMOS のような多面 IPT に関しては確かに圧倒的に優位であり、中央に立った場合に水平視野角は 4 面および 5 面で 270 [deg.], 6 面の COSMOS では全周である。垂直視野角に関しては 4 面の CAVE で上方向 30~40 [deg.], 下方向は 135 [deg.] 前後である。これに対し、HMD では水平方向 40~60 [deg.], 垂直方向 30~45 [deg.] がせいぜいである。ImmersaDesk, Responsive Work Bench などの単面 IPT については、操作者の位置に大きく依存するが、スクリーン面からの距離が 1 [m] の位置からスクリーンサイズ 82.5 [inch] の ImmersaDesk を見た場合、水平方向 80 [deg.], 垂直方向 60 [deg.] となる。

次に、解像度に関してであるが、現在一般的な IPT で使用される解像度は SXGA (1280 × 1024) ないしは 1000 × 1000 であり、対する HMD は VGA (640 × 480) から SVGA (800 × 600) である。これだけを比較すると IPT の方が解像度が高いように見えるが、1 面でカバーする視野角の大きさを考えると実は同程度である。例えば、CAVE の場合、スクリーン形状が正方形のためグラフィックスワークステーションが出力する画面のうち左右端は切り落とされ、中央の約 1000 × 1000 ドットの領域が使用される。CAVE 中央に立った場合、この解像度で視野角 90 度をカバーするため、換算視力は 0.19 程度となる。一方、最近の HMD では SVGA クラスの解像度を有しこれで 40 度から 60 度の視野角をカバーするため、換算視力は 0.33 から 0.22 となり、むしろ CAVE よりも高くなっている。従って、一概に IPT の方が解像度の面で有利であるとは言えない。ただし、IPT の類型の中でも近年人間の視力限界を目指して開発が進められている、1 面に対して複数のプロジェクタを配置した超高解像度システムでは換算視力を 2 倍以上にすることができる。

本章では上記の 2 点以外の、第 3 の特徴に注目する。それは、操作者の頭部運動に対する提示画像の挙動であり、この点に関して次の第 4.3 節で詳細に述べる。

一方、HMD の利点は、まず第一にそのパーソナル性および物理的に占有する空間の小ささにある。IPT のようなディスプレイシステムは部屋ごと作り込む、まさに「建設する」という規模のものであるのに対し、HMD は個人個人が装着するものであり、視覚提示のために特別な、あるいは余分な空間を必要としない。我が国のように空間的コストが極めて高価な環境においては、HMD は情報提示の手段として有効なデバイスになりうる事が期待される。さらに進んで、HMD は携帯も可能であるので、ウェアラブル・コンピューティングなどの用途にも適している。対する IPT は、スクリーンや画像再生デバイスを頭部搭載とせず周囲の空間に固定することにより高性能のデバイスが利用可能となり、前述の広視野化とそれに伴って必要となる解像度や明るさを確保している。

HMD はデバイスを頭部に搭載し眼の直前で外界から与える視覚情報のすり替えを行うため、身体その他の部分との干渉がほとんど生じることなく任意の視覚情報を提示することができる。視覚提示において遮蔽関係の矛盾は奥行き知覚に無視できない影響を及ぼすが、HMD を利用すると身体が映像を隠すことがない。このため、特に近距離に存在する物体の提示には適していると考えられる。しかし、逆に遠方の物体や環境を提示する場合、自分自身の身体を VR 空間において再構成し提示しない限り、逆の遮蔽関係の矛盾、すなわち手を物体と目との間に持っていた場合などに身体で遮蔽されるはずの世界が遮蔽されず、自己投射性の欠如を生じる。この問題に対し、自己発光型ディスプレイデバイスあるいは画像表示スクリーンをバーチャル物体の周囲に偏在させることにより身体と表示物体との遮蔽問題を解決する、オブジェクト指向型ディスプレイ [26] [24] が提案されているが、これはその名の通り物体の提示と直感的なハンドリングインタフェースの共存を目的としたものであり、HMD や IPT のような環境の汎用的な提示を目指したものではない。

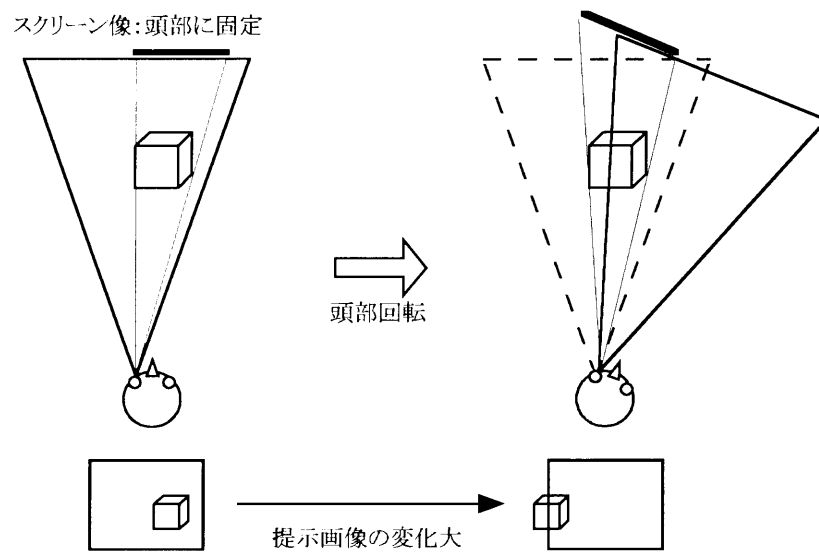
さらにもう一点、HMD では操作者の視点と画像提示を行うスクリーン像との位置関係が固定であるため、提示する画像は一定の透視変換を用いて生成すればよい。これは、コンピュータグラフィックス画像を提示する場合には投影変換指定の更新が不必要であり、カメラにより実世界の映像を提示する場合には光学系固定のカメラが利用可能であることを意味する。この点に関しては第 4.4 節で詳細に述べる。

4.3 操作者の頭部運動に対する提示画像の挙動

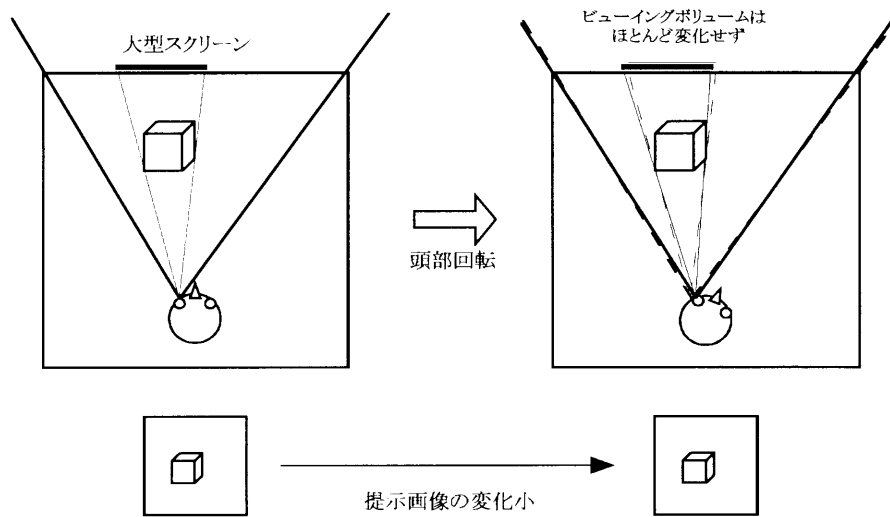
HMD の場合は、スクリーンが空間に対してではなく操作者の頭部に対して固定されているため、操作者の頭部回転によりビューイングボリュームの位置が大幅に変化し、その結果画像が大きく流れることになる (図 4.6 (a))。操作者の運動から画像提示までの遅延が存在すると、HMD では網膜上の像の流れが操作者の頭部運動による内部感覚と矛盾する状況が発生し、これが操作者の感覚へ悪影響を及ぼしていると考えられる。低フレームレート画像、例えば 10 frames per second (fps) 程度の画像では世界が連続に感じられず、また、20fps 程度の画像では世界がぶれて (振動して) いるような感覚が生じる。さらに、遅延の存在するシステムでは世界全体が遅れてついてくるような印象を与える。

これに対し、IPT ではスクリーンが操作者の頭部に対してではなく周囲の空間に対して固定されているため、スクリーンに投影するグラフィックス画像の生成に必要な情報はスクリーンに対する視点の位置のみである。従って、IPT による視覚提示では操作者の頭部回転運動を検出しそれを反映したグラフィックス画像を生成する必要がなく (厳密には両眼立体視を行う場合に左右画像の視差を行うための調整が必要である)、回転運動への対応は実際の空間がまさにリアルタイムに行っていると解釈される (図 4.6 (b))。

CAVE の設計者である Cruz-Neira ら [6] は、トラッキング誤差による提示画像への影響を、頭部並進運動と回転運動 (頭部回転中心から目の位置までのずれにより生じる並進運動を含む) について、

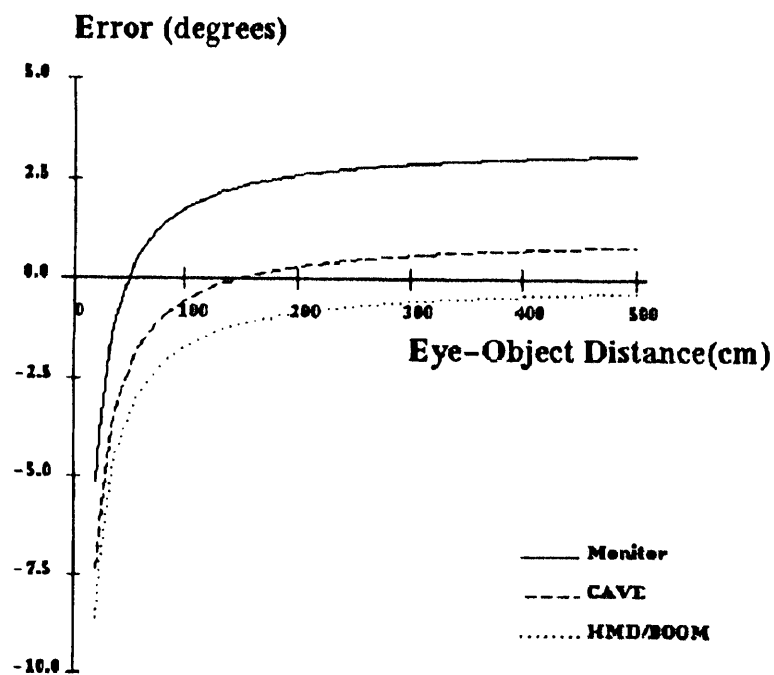


(a) HMD の場合

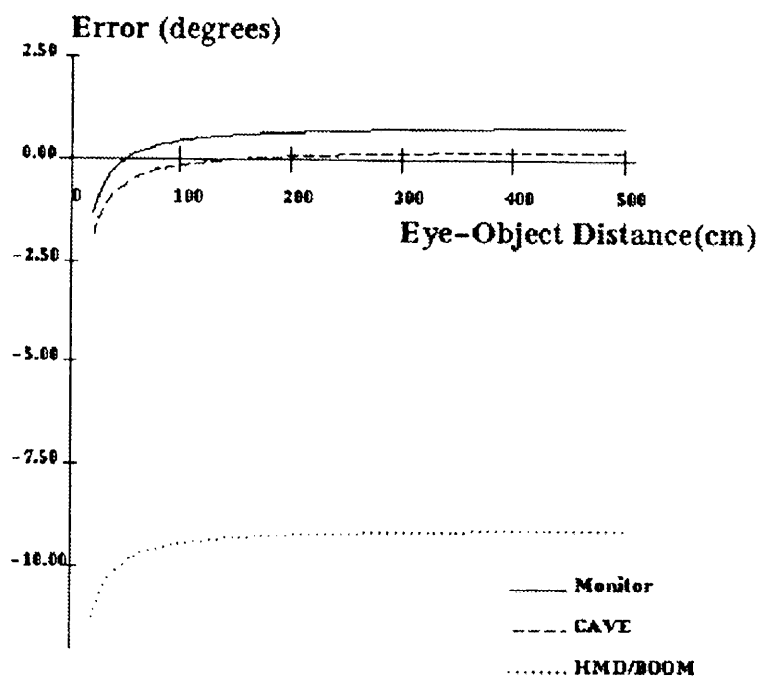


(b) IPT 型視覚提示装置の場合

図 4.6 操作者の頭部回転運動に対する提示画像の挙動



(a) 視点がスクリーンに対して平行移動した場合



(b) 頭部の回転運動が生じた場合

図 4.7: システムの遅延もしくは計測誤差に対する提示物体の角度誤差: CRT モニタ, HMD/BOOM, CAVE の比較 (Cruz-Neira et al. 1993 [6] より引用)

HMD, CRT および CAVE 方式それぞれにおいて、頭部運動のトラッキング誤差が存在する場合にスクリーン上に提示される画像の角度誤差を算出し、IPT の利点を主張している。

図 4.7 において、(a) は操作者の視点がグラフィックス画像生成の計算に使用されている計測位置とずれている場合に、画像上の点を操作者の目の位置から見る方向にどれだけ誤差が生じるかを角度に換算したものである。図において横軸は提示するバーチャル物体と操作者との距離であり、通常の計算機端末で使用される CRT モニタ、HMD、CAVE のそれぞれについて物体の方角の誤差が計算されている。計測誤差あるいは遅延に起因する視点位置のずれとして、3 [cm] という値を用いている。この解析において、CRT モニタと CAVE との違いは画像提示面までの距離であり、CRT モニタは視点からの距離が 0.5m と近いため、相対的に視点移動の影響が大きくなっているものの、全体としての性質は全く同じである。頭部並進運動に対する CAVE と HMD との比較では、CAVE において視点と提示されるバーチャル物体との距離がスクリーンまでの距離（図では距離 1.5m）とほぼ等しい場合に誤差がほとんど生じなくなる点を除いては CAVE 方式の絶対的な優位性を見出すことはできない。

むしろ、特徴的なのは頭部回転運動を伴う場合である。(b) は、ユーザの自然な頭部回転運動（回転中心から目の位置のオフセットによる並進移動も含む）が 90 [deg./s] で行われている時に毎秒 10 フレームで画像の更新が行われている場合に生じる視点位置誤差に起因する提示画像の角度誤差を示すものである。この場合、HMD がトラッキング誤差あるいは遅延相当分を直接的に反映する一定の角度誤差を常に伴うのに対して、CRT や CAVE ではその影響がなく、提示画像の角度誤差が小さく抑えられる結果が得られている。

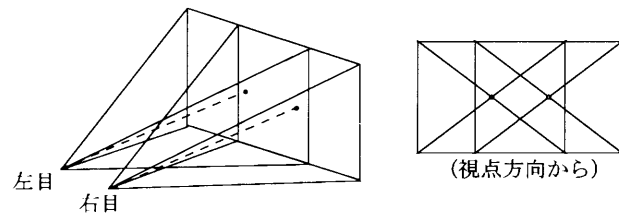
Curz-Neira らの解析はここでとどまっているが、ここでは (b) の結果をもう少し詳しく分析してみる。(b) の計算には頭部回転中心からに対する目の位置のオフセット (5 [cm] という値が用いられている) に起因する平行移動による要素と回転による要素との複合であり、視点位置が全く移動しない純粋な回転に対しては CRT および CAVE は何ら誤差を生じない。つまり、(b) のグラフはオフセットによる平行移動量、すなわち 0.1 秒の間に角速度 90 [deg./s] で半径 5 [cm] の円周上の点が移動する量である 0.8 [cm] という値を (a) と同じ計算に適用した場合のグラフに、HMD だけ 9 [deg] という量の誤差を加えたものになっている。この結果からわかるように、ユーザの自然な頭部回転に対する物体方向の誤差は純粋な回転による寄与が支配的であり、これは画像提示面が頭部に固定されているか、あるいは周囲の空間に固定されているかという違いによって生じるものである。

このように、IPT が高度な視覚提示品質を実現していることの本質は、周囲の空間に対して固定されたスクリーンを用いることにより、操作者の頭部運動、特に回転運動に対して安定した画像を提示可能なことであることが示される。

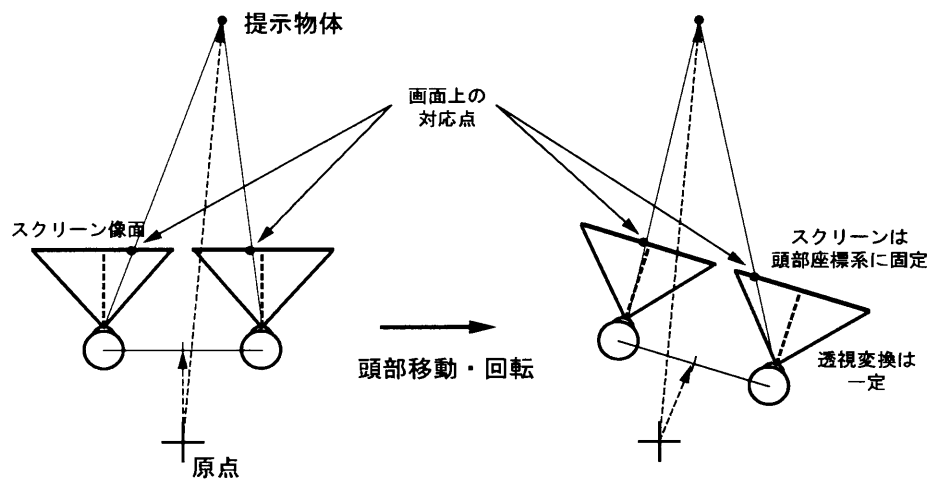
4.4 正確な立体視のための透視変換

両眼輻輳角と視差を手がかりとして正確な立体視を行う場合、画像の生成系と提示系の対応付けを適切に行うことは重要である。すなわち、撮像系（画像生成系）と提示系は対にして設計・運用されるべきであり、撮像系と提示系においてそれぞれ視点と撮像面もしくは提示面とで構成される透視変換におけるビューイングボリューム（四角錐）の形状を一致させることが必要である。このビューイングボリュームの形状およびその頭部運動に対する挙動は視覚提示装置の種類により異なる。

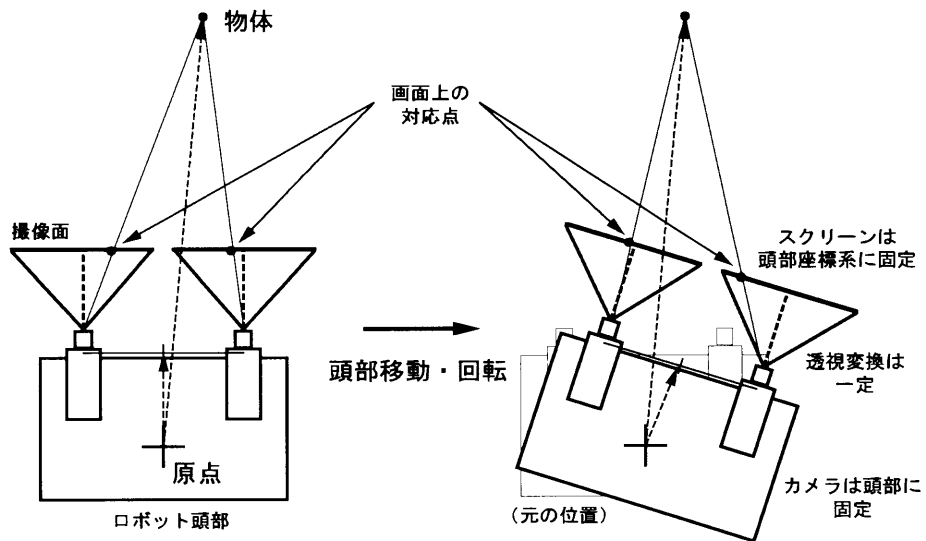
図 4.8 (a) に典型的な HMD を使用した場合のビューイングボリュームの形状、(b) にその頭部運動への対応を示す。最も一般的な HMD では、スクリーン像が前額面に対して平行で、目からスクリーンに下した垂線の足がスクリーン中央となるように設計されている（無限遠点に対して 100% オーバーラップ）。この場合、光軸に対して上下左右対称な透視変換を用いるのが適当であり、画像生成



(a) ビューイングボリュームの形状

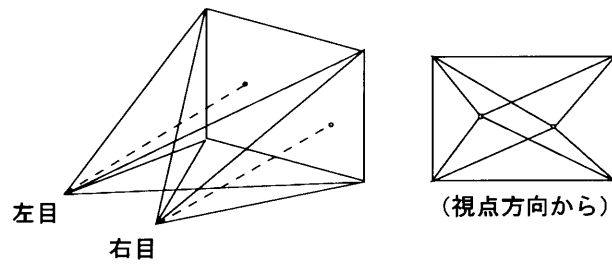


(b) 提示系における頭部運動への対応

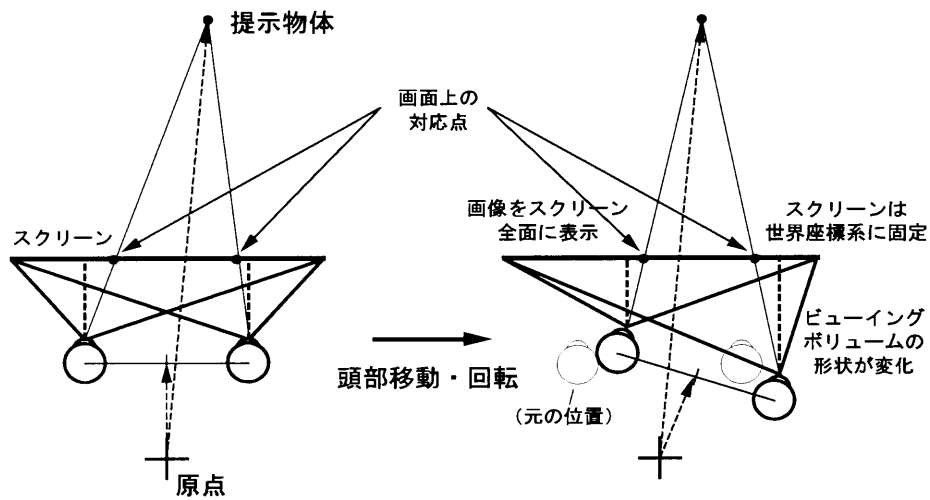


(c) 撮像系における頭部運動への対応

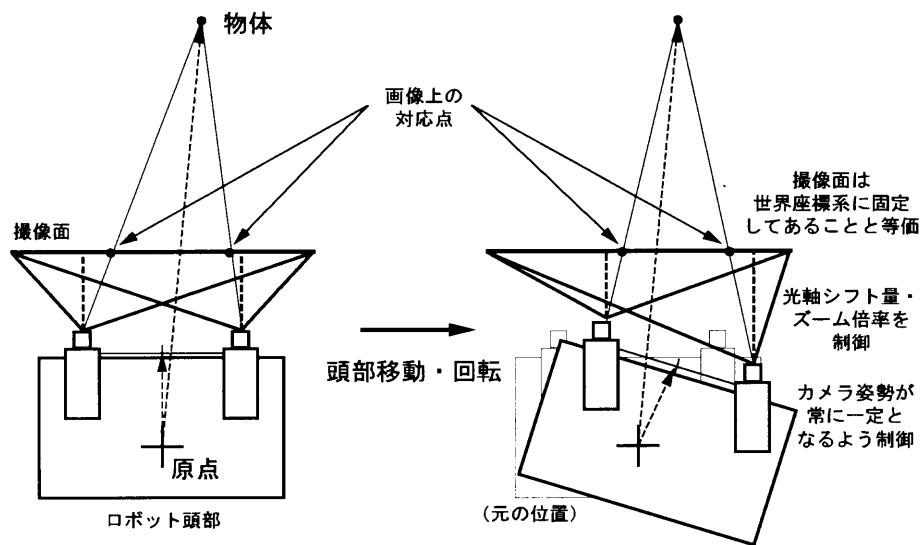
図 4.8 HMD を使用したシステムの透視変換



(a) ビューイングボリュームの形状



(b) 提示系における頭部運動への対応



(c) 撮像系における頭部運動への対応

図 4.9 スクリーン固定型システムの透視変換 (CG 画像提示用)

に必要なパラメータは左右の視点間距離と視野角のみである．HMD の場合はスクリーンが頭部に対して固定されているので，操作者が頭部運動を行ったとしても目とスクリーンとの位置関係は変化せず，従って撮像側の投影変換は常に一定でよい．このようなビューイングボリュームの形状と頭部運動に対する性質を考えると，バーチャル環境を提示する場合には常に一定の上下左右対称な透視変換を用いればよく，また実環境へのテレイグジスタンスにおいては普通のカメラを用いることができる．さらに，頭部運動を行った時に左右それぞれの視点・スクリーンの組と遠隔/バーチャル環境との位置関係が等しく変化するので，図 4.8 (c) のように左右のカメラの位置関係を固定し（通常はロボット頭部に眼間距離分だけ離して平行に配置して）それを操作者の運動に追従するよう制御すればよい．

一方，IPT などの固定スクリーン型システムにおいて，コンピュータグラフィックス画像の提示に用いられているビューイングボリュームの形状と頭部運動への対応をそれぞれ図 4.9 (a), (b) に示す．ビューイングボリュームは左右それぞれの視点を頂点としスクリーン面（あるいは画像表示ウィンドウ）を底面とする四角錐であり，この場合は頂点から底面へ下した垂線の足は一般にスクリーンの中央ではない．言い換えれば，視点と視野中心とを結ぶ直線が投影面に対して斜めになっている状況である．また，操作者の頭部運動により目とスクリーンの位置関係が実時間で変化するため，上記の四角錐の形状を実時間で更新する必要がある．

	HMD	IPT
頭部運動に対する 画像安定性	×	○
	第5章で改良	
VR空間提示	○	○
遠隔実空間提示	○	×
		第6章で可能に

図 4.10 HMD と IPT の長所・短所と適用分野

4.5 まとめ

以上のように，VR・テレイグジスタンスにおける代表的な視覚提示デバイスである HMD と IPT（スクリーン固定型システム）は，それぞれ利点・欠点を持つ（図 4.10）．HMD を利用したシステムでは頭部運動，特に回転運動に対する提示画像のオプティカルフローについて問題が生じやすい．すなわち，CG 画像提示や低速回線を経由する遠隔画像提示を行う時，「世界揺れ」の問題がつきまとうが，これに対する解決策を第 5 章において提案する．一方，IPT は人間の頭部運動に対して安定した世界の提示が可能であるが，時変・非対称透視変換が必要であるため，実写画像をそのまま利用するリアルタイムテレイグジスタンスには適用できなかった．これを解決するのが第 6 章の方法である．

第 5 章

HMD 利用時の頭部回転運動に対する画像安定化

5.1 概要

VR (Virtual Reality, 人工現実感) システムにおける視覚提示の品質を向上させるには、フォトリアリスティックな画像の生成などの空間的な品質と、リアルタイム性の追求などの時間的な品質の双方から探求を行うことが重要である。空間的な品質の向上に関しては、近年映画産業においてコンピュータグラフィックス映像が積極的に導入されている状況に代表されるように、実写映像と見まがうばかりの画像の生成が可能になってきている。しかし、これらの手法は一般に非常に複雑な計算を必要とし、VR システムのような実時間性を要求する状況においては適用できない。

VR システムにおいては、操作者の行動に対して実時間でバーチャル空間の状態を更新し、さらに操作者へ提示される感覚信号へ反映することがきわめて重要である。しかし、現在の多くのシステムでは、バーチャル環境画像提示の時間的な品質は必ずしも十分とは言えないレベルに留まっている。

時間的な品質の制約となる一つの要因として、多くのグラフィックスシステムに用いられているダブルバッファ法に起因するフレームレートの問題が存在する。現在の VR システムでは、多くの場合 10Hz ~ 20Hz のフレームレートで画像更新が行われているが、この値は人間にとって世界が時間的に連続であると感じられるために十分な値ではない。近年コンピュータグラフィックス生成のためのハードウェアは急速な進歩を遂げているが、グラフィックスシステムの性能が向上しても、より複雑な環境や精密なモデルに基づいた画像生成が行われるため、フレームレートは必ずしも高速化されず、ほぼ一定にとどまる傾向が見受けられる。さらに時間的な品質の制約となるもう一つの要因として、操作者の運動や操作から視覚情報提示までの総合的な遅延時間の問題 [35] が挙げられる。VR システムにおいては、計測デバイスの特性やグラフィックス画像生成、さらには VR システムを構成する要素間の通信のため、必ず遅延が発生する。低いフレームレートや大きな時間遅延は、HMD (Head-Mounted Display, 頭部搭載型画像提示装置) や全天周型スクリーンを利用する場合など、没入感の高い視覚提示を行う場合には操作者に対してストレスを感じさせる原因になっていると考えられる。

そこで本研究では、グラフィックス画像生成速度の向上とは独立に、等価的にフレームレートの向上と遅延時間の低減を行う手法を提案する。すなわち、提示される画像のオブティカルフローに着目し、離散的にのみ入手可能な画像と補助的なセンサ情報を元に、操作者の頭部運動に対応して適切なオブティカルフローを持つ画像をリアルタイムで操作者に提示するという手法を提案する。

関連分野としては、ハードウェア性能の限界に起因するフレームレートの問題への対策を目的として、米国ノースカロライナ大学 (UNC) の研究グループにより Frameless Rendering の研究 [3] が行われているが、これは画面上のピクセルをランダムに選択し逐次更新する手法であり、画面全体のオブティカルフローを正確に再現することはできない。

また、近年同じ機関の研究グループにより、実写画像に基づいた VR 空間の画像生成 (Image-Based Rendering) の一手法として Post-Rendering Image Warping という技術 [33] が開発されている。こ

れは、高品質なグラフィックス画像が得られる代わりに 1 枚の画像を生成するために膨大な計算量と長時間を要するレイ・トレーシング型の画像生成アルゴリズムと併用して効果を発揮する技術であり、ある視点を基準に生成された二次元的な画像を元に別の視点からの画像を生成するアルゴリズムである。この場合、視点の変更による画像変換は画像自体の生成と比較して効率の良い手法で準リアルタイムに行われ、写実的な高品質画像をほぼ実時間で動かすことが可能になっている。しかしながら、この技術をもってしても、HMD などを利用する場合には人間の感覚知覚特性レベルでの実時間 (10ms オーダー) で提示される画像を更新することが必要であり、こうした処理ではこのレベルの実時間性を満足されることは困難である。

一方、HMD を用いた視覚提示システムにおける遅延への対処という観点からは、主に実世界と CG 画像とを重ね合わせて提示する Augmented Reality [2] の分野でいくつかの手法が研究されている。例えば、江村ら [8] [9] は磁気センサにより得られる低速ではあるが絶対位置・姿勢に関する計測情報と、レートジャイロにより得られる角速度情報をセンサフュージョンにより統合して良質な計測情報を得る研究を行った。また、横小路ら [86] は加速度センサと画像によりシステム遅延分の予測を行い VR 画像と実世界画像とを精度良く重ね合わせることに成功している。これらの研究は計測情報、すなわちシステムを伝達する情報経路の入口の部分での遅延補償を目指したものであるが、これに対し本章で述べる手法は、システムの出口、すなわち最終的な画像提示段階において頭部運動感覚と提示される視覚情報との整合性を確保するという思想に基づくものである。

オブティカルフローを適切に提示することは、操作者の頭部運動感覚と視覚との融合を高いレベルで実現し、操作者の負担軽減をもたらすことが期待される。また、本手法は低速回線を経由するテレイグジスタンス作業など、入手できる環境画像の高速な更新が不可能な場合にも適用可能である。本章では、HMD 装着時の頭部回転運動に起因する均一なオブティカルフローに関して、高速なセンサ情報を用いて適切に補正を行い提示する手法を提案する。

5.2 回転運動に関する HMD 提示画像の補正手法

5.2.1 基本原理

VR システムにおけるフレームレートおよび遅延の問題に対処するためには、グラフィックスハードウェアを高速化して画像生成に伴う遅延を少なくするというアプローチも当然考えられるが、前述のようにグラフィックスシステムの高速化は必ずしもフレームレートの向上につながっていない。加えて、図 5.1 に示すように、運動の発生から画像提示に至るまでの間に、計測システム、通信チャンネル、およびグラフィックス生成という遅延要素が存在し、これらが直列に接続されているため遅延の累積は避けられない。このため、グラフィックスハードウェアのみを高速化しても、動作計測やその他の原因による遅延は解消されず、結局は一定の遅延が残ってしまうことになる。

本研究では、このような問題に対処するため、以下のようなアプローチをとる。

- 情報の流れの分割と階層化：従来型 VR システムは、前述のように情報の流れが単一のループで構成され、その経路の随所に遅延発生要素と情報更新レートのボトルネックが存在することに問題があった。これを解消するため、従来の情報ループに加えて、高速に情報更新を行うローカルループを構成する。VR システムにおける実時間性は操作者の感覚知覚という文脈でとらえられるべきであり、その意味でローカルループは操作者に近い場所で構成されるのが適当である。こうしたループ多重化はロボット制御の分野では日常的に用いられている手法であり、そうした事例からも実時間性を重視する状況においては有効な手段であると考えられる。

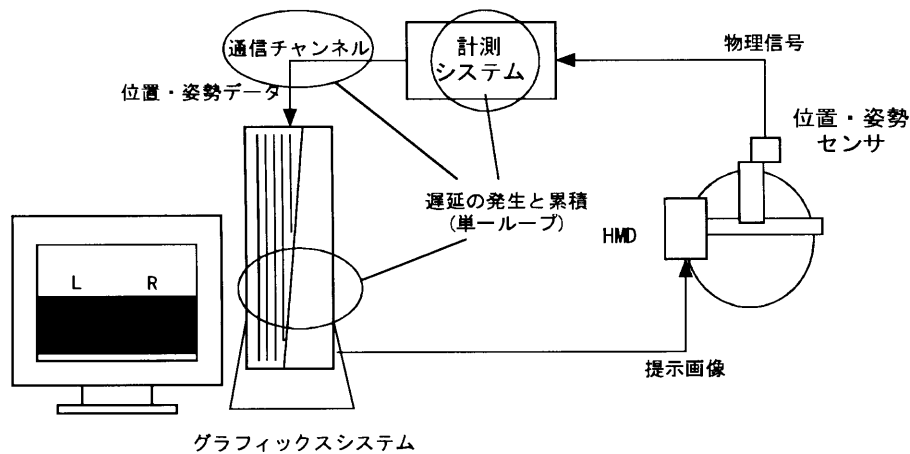


図 5.1 従来型 VR システムの構成

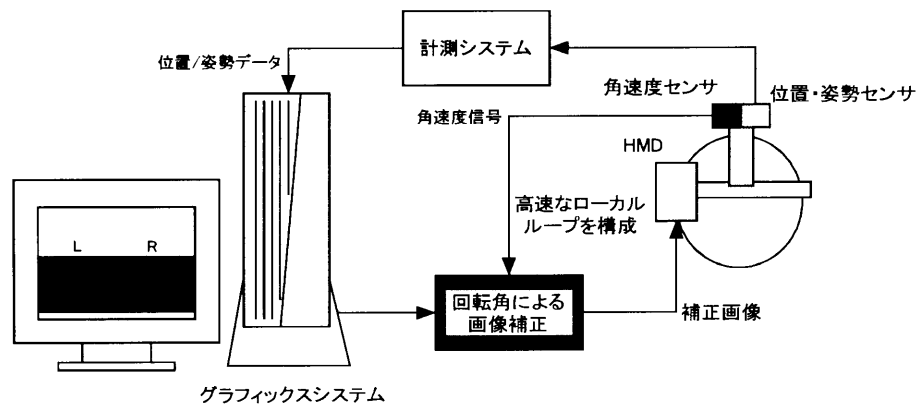


図 5.2 回転角補正型 HMD システムの概念

- 適切な質を持つ情報の入手：従来型 VR システムのもう一つの問題点として、操作者の運動計測を Polhemus 社の磁気センサのようなパッケージ化された位置・姿勢センサに頼っていることが挙げられる。このようなセンサは単体で必要な情報を得られるため非常に便利であるが、安定性を向上させるために信号処理ユニット内で時間的な平均化処理が行われるのが普通である。このため、計測システム自体がある程度の遅延を発生させてしまっている。本研究においては、上記の高速情報ループに用いる運動情報として、「動き」そのものをダイレクトに用いる。すなわち、絶対的な角度情報ではなく、角速度情報を用いることにより高速ループに対応する。角速度情報を計測するセンサとしては、普及品のレートジャイロを使用可能であり、コスト的にも負担にならない。

以上のような点に着目し、操作者の頭部回転運動に起因する提示画像の遅れをキャンセルする機構を導入することにより、HMD による視覚提示方式の弱点の一部を解消する手法を提案する。本研究で提案するシステム構成を図 5.2 に示す。本手法は頭部回転運動により発生する画像全体の均一なオプティカルフローを的確に提示するものであり、人間の前庭動眼反射 (VOR) のメカニズムに適合する視覚提示を行うことも狙っている。

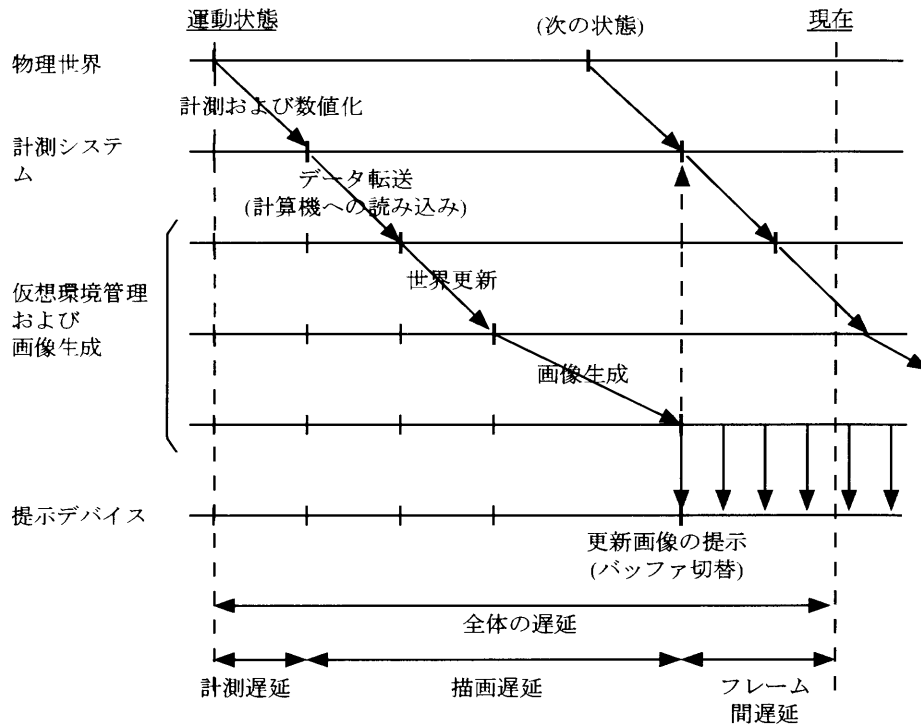


図 5.3 VR システムにおける提示画像の遅延

5.2.2 提示画像の遅延の分析

操作者に提示される画像は操作者の運動状態を反映しているが、操作者に提示が行われるまでの間に様々な遅延要素を含むため、画像が提示された時点では既に過去の状態を反映している。また、1枚のグラフィックス画像が提示されてから次の画像に更新されるまでの間はさらに遅れが重なることとなる。この様子を図5.3に示す。

動作発生から提示画像までの遅延をその性質により分類し、以下のように呼ぶことにする。

- 描画遅延：グラフィックスシステムでのループ時間。データ読み込み、世界計算、グラフィックス生成を含む。
- 計測遅延：計測システムがデータを用意するまでの時間など、画像生成に使用される状態の時間から画像生成ループが始まるまでの時間の合計。
- フレーム間遅延：最新の画像が提示されてから現在までの間の経過時間。

現在操作者に提示されている画像の遅延は、これら3種類の遅延の合計となる。

5.2.3 回転角補正量の算出法

頭部回転運動の検出には、高速なジャイロを用いることを想定する。ジャイロは角速度センサであり、角度の絶対値を求めるには積分演算が必要である。このとき、ドリフトや積分誤差による誤差が蓄積するため、ジャイロによる計測データを積分した絶対角度情報として利用することは不利である。そこで、ジャイロの高速性を生かしつつ誤差の蓄積を回避するため、ジャイロが出力する角速度の短時間積分値のみを用い、画像が更新された時点で過去の積分値を破棄する手法を考案した。角度の絶

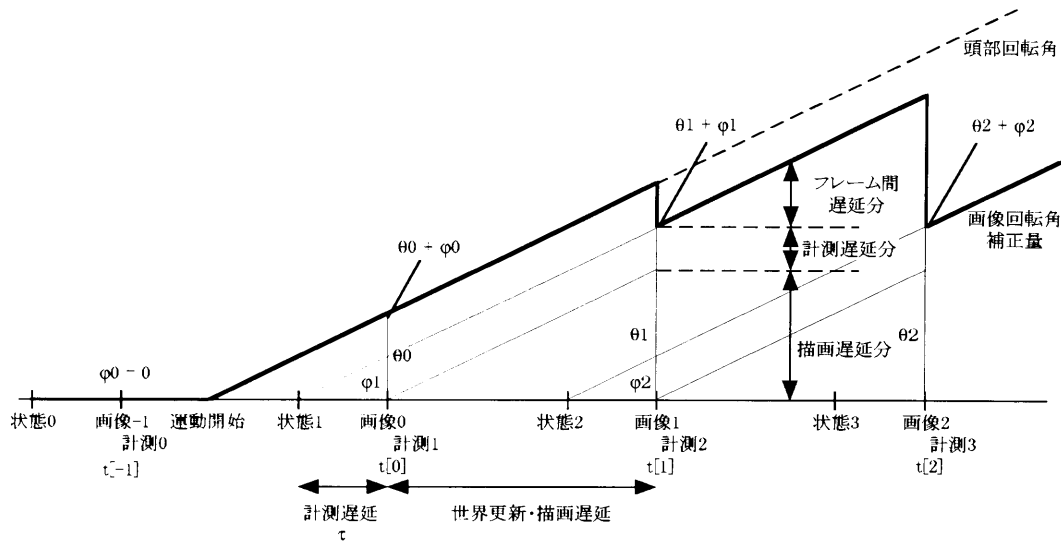


図 5.4 頭部回転角補償量

対値に関する情報は通常の頭部位置・姿勢センサにより計測されており、ある遅延時間を伴ってグラフィックス画像に反映されるので、ここでは短時間の角度変化分のみを補償すればよい。

図 5.4 に頭部回転角の補正量算出の方法を示す。全補正量は、現在表示されている画像の計測遅延分、描画遅延分、およびフレーム間遅延分の合計である。これらのうち、計測遅延分と描画遅延分は一枚の画像が提示されている間は定数であり、フレーム間遅延分は現在の頭部回転角速度の積分値により実時間で変化する。

5.2.4 実装手法の検討

画像提示の遅延を補償するため、グラフィックスハードウェアが生成した画像を、回転角の差を補正して画像生成の際に参照した頭部姿勢に対して正しい位置に提示する。これを行うための手段としては、以下のような選択肢が考えられる。

- (a) 2次元画像のシフトと回転による実装：単純な実装であり、ローカルなハードウェアでの処理に適する。グラフィックス生成に使用された運動状態の時点（運動時点と呼ぶ）と提示時点のビューイングボリュームの形が異なるので、特に周辺部において画像の歪みを伴う。しかしながら、ローカルハードウェア指向と処理の単純化により画像更新を高速に行うことが可能であると考えられる。
- (b) 2次元画像のシフト・回転＋伸縮による実装：周辺部の画像歪みを低減するため、単純なシフトおよび回転に加えて、画像全体の伸縮を行う。画像全体を均一に拡大・縮小した場合はやはり画像歪みが生じるが、視野周辺部の誤差を小さく抑えることができる。非均一の拡大・縮小は、正確に行うと次のテキスチャマッピングと等価になる。
- (c) テキスチャマッピングによる実装：運動時点と現在の頭部方向・視点位置を比較し、現在の視点位置から見てスクリーンが運動時点のそれと一致するような三次元空間位置に提示を行う。この手法では原理的に画像の歪みは存在せず IPT 型提示装置と同様の条件となる。これを行うに

はテクスチャマッピングが有効であるが、テクスチャマッピングは高価なグラフィックスハードウェアを必要とし、かつリアルタイムでビデオ画像を取り込みテクスチャメモリへ転送するハードウェアが必要になる。また、処理をグラフィックスコンピュータで行うことになり、計測情報をコンピュータに読み込むための変換/通信オーバーヘッドが伴う。

以上の方法を図 5.5 に示す。それぞれの図において、記号 φ は画像上の点が正中線方向となす角、 θ は画像生成に用いられた運動状態から現在の運動状態までの頭部回転角を表す。HMD のスクリーンは頭部に固定されているため、画像が提示される時点では角度 θ だけ回転しているが、それを等価的に画像生成時点の位置に戻すことにより、操作者にとって周囲の世界が揺れることなく安定して存在しているように感じられる。

提案する三種類の手法のうち、手法 (a) の場合はグラフィックスシステムによる画像生成部分には手を加えず、HMD 本体あるいはコンピュータと HMD の間に処理装置を挿入するという実装形態が可能になる。すなわち、図 5.2 に示したシステムの概念とは最も整合性が高い。この場合、さらに画像処理の方式に関して以下のような実装の選択肢が考えられる。その際、ビデオカメラの手ぶれ補正方式と対比させると興味深い。

- デジタル/電子的実装：画像を一旦メモリに取り込み、DSP などでの電子的処理を行う。A/D、D/A の手間が増え、ビデオレートのオーダーで遅延を挿入することになる（遅延量に加えておくことにより、補正は可能）が、自由度は最も高い。手ぶれ補正では「イメージエリア方式」に対応する。
- アナログ/電気的実装：走査線の原点調整などの手段により、画像提示デバイス上の画像位置を制御する。この手法では画像の回転には対応できず、yaw および pitch 方向のみの補正となる。
- アナログ/光学的+機械的実装：HMD スクリーン像の眼球に対する位置と方向を光学的に制御する。家庭用ビデオではジャイロとアクティブプリズムにより手ぶれ補正を行っており、「光学式（アクティブプリズム方式）」に対応する。

本研究では、できる限りローカルな処理が可能な実装方法を模索するため、以下では単純画像シフトによる手法の解析を行う。

5.3 画像歪みの見積

単純画像シフトにより提示画像遅延を補正した場合、理想的な場合（テクスチャマッピングによる実装）と比べてビューイングボリュームの形状が変形するので、画像上の点の位置誤差が発生する。

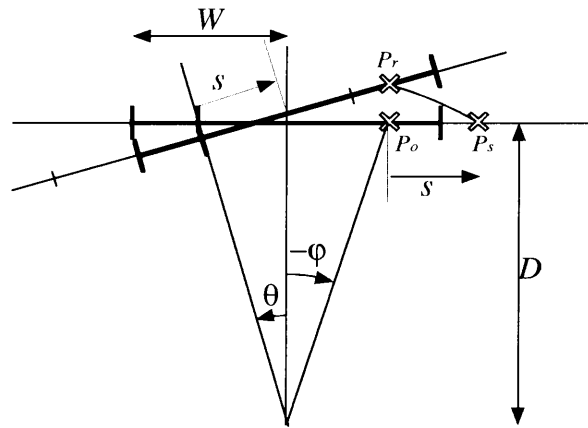
頭部が角度 θ だけ回転した場合の画像シフトは、画像中心が元の位置に一致するように行うものとする。 P_o は元の画像上の点の位置、 P_s はそれを画像シフト量 s だけシフトさせた点、 P_r は視点を中心として P_s を角度 θ 回転させた点、 P_p は P_r をスクリーン上へ投影した点である。 P_o の座標を

$$P_o = \begin{bmatrix} x \\ D \end{bmatrix} \quad (5.1)$$

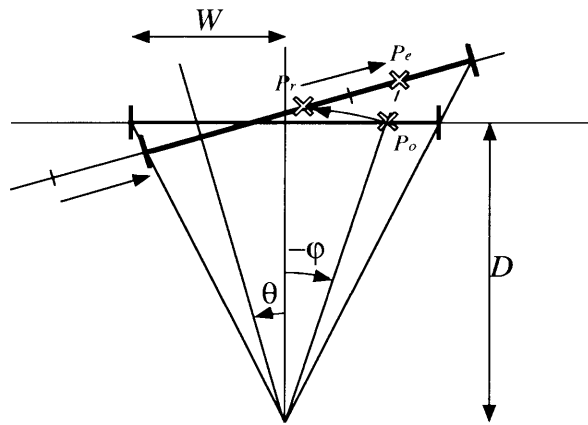
と定め、この点の視野中心からの見込み角を θ とする。画像中心を一致させるためのシフト量 s は

$$s = D \tan \theta \quad (5.2)$$

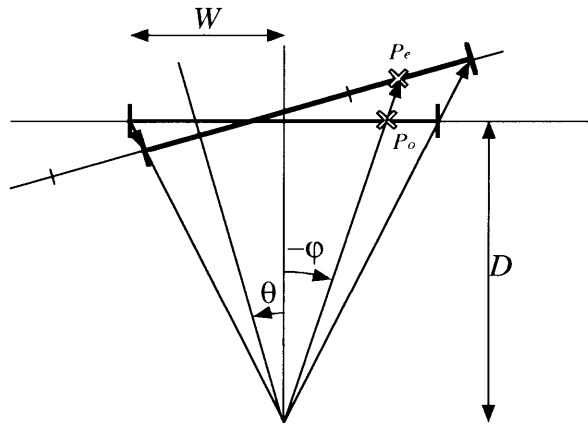
である。



(a) 二次元画像の単純シフト方式



(b) 二次元画像のシフト+伸長圧縮方式



(c) テキスチャマッピング方式

図 5.5 画像補正方式

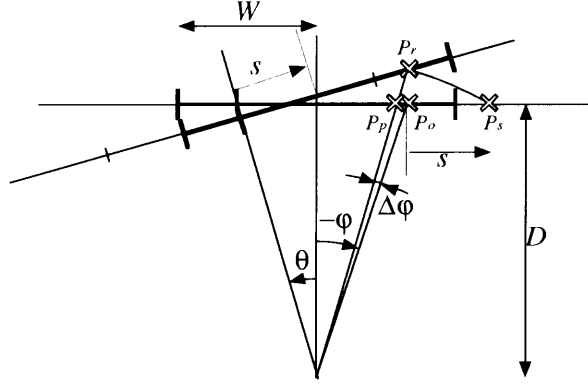


図 5.6 画像シフトによる提示画像の歪み（角度誤差）

5.3.1 角度誤差による見積

提示画像の角度誤差を図 5.6 に示す．正しい位置の画像の振れ角を φ とすると，この点の x 座標は

$$x = -D \tan \varphi \quad (5.3)$$

であるから，シフトした点 P_s の x 座標，振れ角はそれぞれ

$$x_s = x + s = -D \tan \varphi + D \tan \theta \quad (5.4)$$

$$\varphi_s = -\tan^{-1} \frac{x_s}{D} = \tan^{-1}(\tan \varphi - \tan \theta) \quad (5.5)$$

である．これを回転させた画像上の点 P_r は

$$\varphi_r = \varphi_s + \theta = \tan^{-1}(\tan \varphi - \tan \theta) + \theta \quad (5.6)$$

となる．よって画像歪みの角度は

$$\Delta\varphi = \varphi_r - \varphi = \tan^{-1}(\tan \varphi - \tan \theta) + \theta - \varphi \quad (5.7)$$

と計算される．補正なしの場合は一律に

$$\Delta\varphi_n = \theta \quad (5.8)$$

である．仮に操作者が 90 [deg/s] の速度で頭部を回転させ，グラフィックスシステムが毎秒 10 フレームで描画を行っているとする，回転角補正量の最大値 θ は（描画遅延とフレーム間遅延を合わせて）18 [deg] である．この場合の角度誤差を図 5.7 に示す．図に示すように，この値は回転前のスクリーン中心（ $\varphi = 0$ ）と回転後のスクリーン中心（ $\varphi = \theta$ ）において 0 となっており，この間の角度誤差が小さく抑えられている．このため，操作者が画像中心付近を注目している場合には有効な補正手法であると考えられる（ $\varphi = 9$ [deg] において，補正なし 18 [deg] に対して補正あり 0.45 [deg]）．

頭部回転に対する画像シフト補正を行わない場合と行った場合の角度誤差の値の比較を表 5.1 に示す．

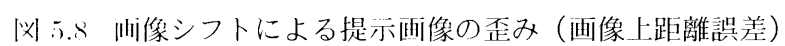
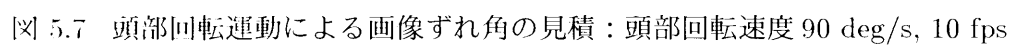


表 5.1 画像ずれ角の見積 (単位 degree)

	補正なし	補正あり	
		中央部	周辺部
視野角 40 度	18	0	3.44
視野角 60 度	18	0	5.94
視野角 90 度	18	0	10.04

5.3.2 画面上距離誤差による見積

画像シフトを行った場合の画像上の点の位置ずれの様子を図 5.8 に示す．元画像上の点 P_o をシフトした点，およびそれを角度 θ 回転させた点 P_s , P_r の座標は

$$P_s = \begin{bmatrix} x + s \\ D \end{bmatrix} = \begin{bmatrix} x + D \tan \theta \\ D \end{bmatrix} \quad (5.9)$$

$$P_r = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} P_s = \begin{bmatrix} x \cos \theta \\ x \sin \theta + \frac{D}{\cos \theta} \end{bmatrix} \quad (5.10)$$

であり，

$$P_p = \frac{D}{x \sin \theta + \frac{D}{\cos \theta}} P_r \quad (5.11)$$

であるからその x 座標は

$$x_p = \frac{D}{x \sin \theta + \frac{D}{\cos \theta}} x \cos \theta \quad (5.12)$$

と表される．よって元の点の位置との画像の位置ずれは

$$\Delta x = x_p - x = -\frac{x D \cos^2 \theta - x^2 \sin \theta \cos \theta - x D}{x \sin \theta \cos \theta + D} = -x \cdot \frac{\frac{x}{D} \sin \theta \cos \theta + \sin^2 \theta}{\frac{x}{D} \sin \theta \cos \theta + 1} \quad (5.13)$$

計算される．これを HMD スクリーンまでの距離 D で割って無次元化すると，

$$\frac{\Delta x}{D} = \frac{x_p - x}{D} = -\frac{x D \cos^2 \theta - x^2 \sin \theta \cos \theta - x D}{x \sin \theta \cos \theta + D} = -\frac{x}{D} \cdot \frac{(x/D) \sin \theta \cos \theta + \sin^2 \theta}{(x/D) \sin \theta \cos \theta + 1} \quad (5.14)$$

となる．この値は $x = 0$ および $x/D = -\tan \theta$ において 0 になり， $-\tan \theta < x/D < 0$ で正，それ以外で負の値をとる．

一方，補正を行わない場合は頭部回転による点 P の移動先は

$$P_{nr} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} P_o = \begin{bmatrix} x \cos \theta - D \sin \theta \\ x \sin \theta + D \cos \theta \end{bmatrix} \quad (5.15)$$

であるから，

$$x_{np} = \frac{D(x \cos \theta - D \sin \theta)}{x \sin \theta + D \cos \theta} = \frac{Dx - D^2 \tan \theta}{x \tan \theta + D} \quad (5.16)$$

よって画像上の点のずれは

$$\Delta x_n = x_{np} - x = -\frac{D^2 \tan \theta + x^2 \tan \theta}{x \tan \theta + D} = -\frac{(D^2 + x^2) \tan \theta}{x \tan \theta + D} \quad (5.17)$$

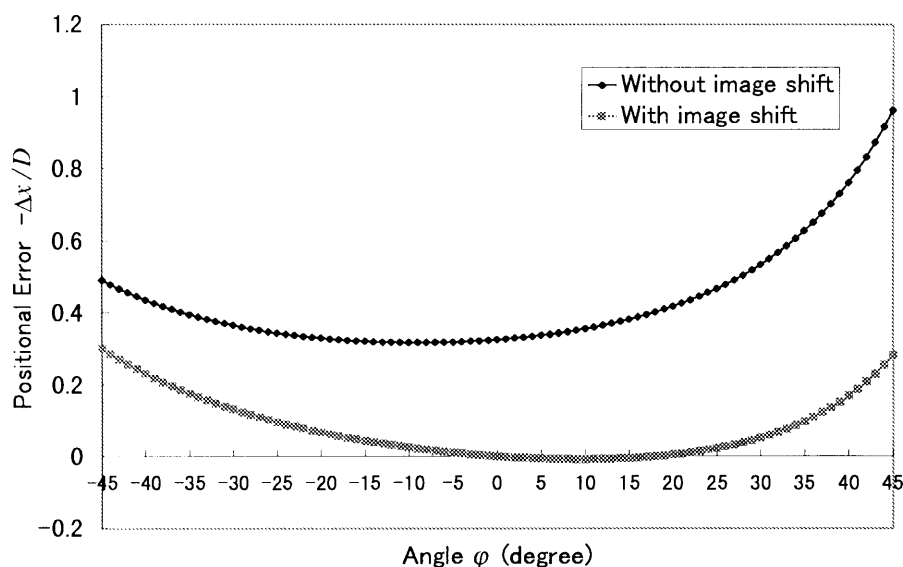


図 5.9: 頭部回転運動による画像位置ずれの見積：頭部回転速度 90 [deg/s], 10 [fps] ($\theta = 18^\circ$, $\tan \theta = 0.32$)

となる． D で割って無次元化すると，

$$\frac{\Delta x_n}{D} = -\frac{(1 + (x/D)^2) \tan \theta}{(x/D) \tan \theta + 1} \quad (5.18)$$

である．このずれは回転方向視野端で最大になる．

頭部回転速度 90 [deg/s]，フレームレート 10 [fps] の場合の画像位置ずれの値を図 5.9 に，最大値の見積りを表 5.2 に示す． $-\tan 18^\circ < x/D < 0$ で正，それ以外で負になっており，視野中央付近において画像歪みが小さく抑えられていることがわかる ($\theta = 9$ [deg] の時，補正なし 0.35 に対して補正あり 0.008)．

表 5.2 画像位置ずれの見積

	補正なし		補正あり	
	中央部	周辺部	中央部	周辺部
水平視野角 40 度	0.325	0.417	0	0.067
水平視野角 60 度	0.325	0.533	0	0.130
水平視野角 90 度	0.325	0.963	0	0.300

5.4 実時間シミュレーションによる検証

本手法の原理を確認するため，グラフィックスワークステーション上でシミュレーションを行った．本研究は低フレームレートのグラフィックス画像を用いて等価的にビデオレートの画像を生成することを目的としているが，ここではまず，逆にビデオレート以上の速度でグラフィックス画像の更新が

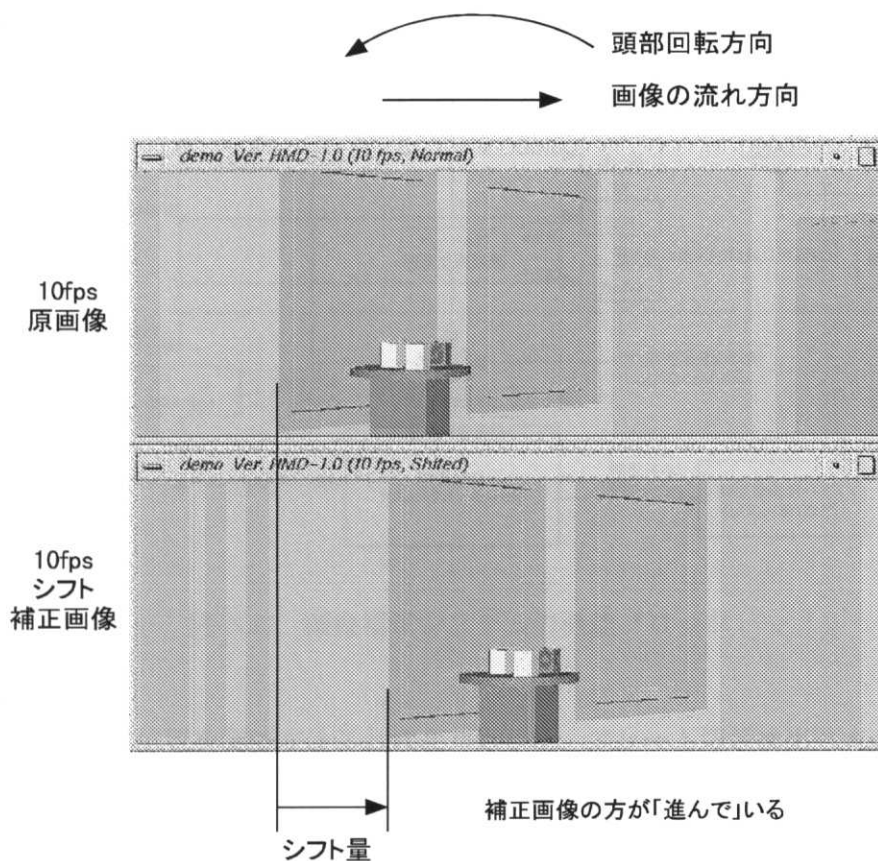


図 5.10 実時間シミュレーションの画像例

可能なシステム (SGI Indigo2 Maximum IMPACT) を用い、低フレームレート画像の生成は適切な待ち時間を挿入することによりシミュレートした。実時間シミュレーションのソフトウェアは C++ 言語で記述しており、Open GL および GLUT [27] を使用し、IRIX 6.2 オペレーティングシステムの上で動作する。また、本ソフトウェアを Windows 95, 98, NT 4.0 上へ移植し、動作を確認した。

シミュレーション画像の例を図 5.10 に示す。この結果、画像シフトによる実装では離散的な原画像を用いることによる歪みは存在するものの、画像全体の流れはビデオレートでの画像更新と遜色ない挙動が確認された。

5.5 試験的実装

実時間シミュレーションの結果に立脚し、グラフィックス画像生成と独立に画像補正を行うシステムの試験的実装を行った。本研究で提案する手法は、本来ならばハードウェア実装を行うことが理想的であるが、シミュレーションの次の段階として、パーソナルコンピュータと市販の画像処理ボードを用い、ソフトウェア的に実装を行った。

5.5.1 システムの機能構成

画像補正システムを実装するための機能ブロックを図 5.11 に示す。これらのうち、回転角の算出などはソフトウェア的に行っている。また、今回の試作では、画像シフトは PC 搭載のグラフィックス

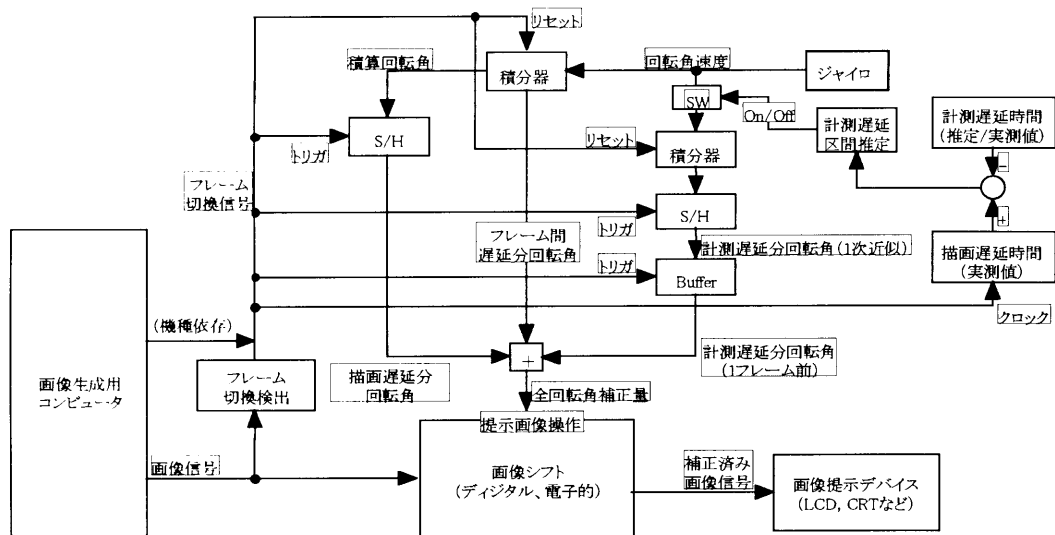


図 5.11 画像補正システムの機能ブロック

ボード上に表示する画像位置を制御することにより行った。フレーム切替検出は本来バーチャル環境画像を生成するコンピュータの側で信号として出力できるようになっていることが望ましいが、そのようなハードウェアは現在一般的ではない。出力された画像からグラフィックスのフレーム切替を算出することは困難を伴う作業であるが、今回はこれを PC に搭載した画像処理ボードにより行った。

5.5.2 フラットフォームの選択とソフトウェアの構造

画像補正システムを構築するにあたっては、高価なグラフィックスワークステーションではなく、安価に入手でき近年進歩の著しいパーソナルコンピュータを選択した。主な構成は、以下の通りである。

- CPU: Pentium-II 450MHz
- メモリ: 128MB SDRAM (100MHz)
- グラフィックス: nVIDIA RIVA TNT-based AGP Graphics Card, SDRAM 16MB
- 画像処理ボード: 富士通製 カラートラッキングビジョン稼働

稼働 OS は、マルチメディア系ボードのドライバの入手しやすさ、またリアルタイム指向のゲーム用ソフトウェア技術を利用しやすいことなどから、Microsoft Windows 98 とした。

本研究の技術的な中核部分をなす画像のリアルタイムシフトは、Microsoft DirectDraw の機能を利用して実装した。PC のシステムメモリおよびグラフィックスカード上のメモリに用意された画像を毎秒 60 回のレートで位置を指定しながら転送し、極めて滑らかな実時間画像シフトが実現できることを確認した。

フレーム切替検出部分は、理想的にはグラフィックス原画像生成側から何らかの形で信号が出力されることが望まれるが、現時点でこの機能を備えた計算機は稀少かつ高価であり、画像補正側で対応することとした。フレーム切替検出には、富士通製 カラートラッキングビジョンボードを用い、画像上複数点の周辺のオブティカルフローを算出し、その平均値が一定の値を超えるとグラフィックス

画像が新しいフレームに切り替わったと判断する．また，同ボードは画像取り込みも可能であり，今回の試験実装では画像取り込みボードとしても利用している．

遅延時間の推定は，グラフィックス画像と実カメラ画像との比較により行った．推定された値を用いて，画像補正システムを稼働させた結果，画像シフト方式による遅延保証機能とフレームレート向上機能が実現されることが確認された．

5.5.3 課題と今後の展望

今回の試験的実装では機能的な実現可能性が確認できたが，性能的にはまだ課題が残されている．すなわち，画像が一旦計算機内のシステムバスとメインメモリを経由する上，入力側と出力側の画像フォーマットが一致していないため画像形式変換を行う必要があり，この段階で余計な遅延を生じている．これは本来ボード上での画像処理用に設計されたものを画像取り込みボードとして兼用していることも一因であり，画像取り込みから出力までを極力短い遅延時間内に行うようシステムを改良する必要がある．

さらに，繰り返し述べてきたが，本研究の最終的な目標は，画像補正機能をハードウェア的に実装することであり，今後ハードウェア的な実装の研究開発を行うことが重要であると考えられる．

5.6 まとめ

オブティカルフロー提示の考え方にに基づき，HMD 装着時の頭部回転運動に起因する提示画像遅れの補償と等価的なフレームレート向上を行う手法を提案し，実時間シミュレーションおよび試験的な実装によってその効果を確認した．本手法は頭部搭載型デバイスに対する汎用的な技術であり，操作者の使用感に直結する時間的品質の向上をもたらすものである．今後本手法がハードウェア的に実装され，安価に利用できるようになれば，パーソナルな情報提示デバイスとしての HMD の利用が加速され，情報化社会における有効な情報提示手段として定着していくことが期待される．

第 6 章

固定スクリーンを利用した実環境へのトレイグジスタンス

6.1 概要

VR における視覚提示システムとして、CAVE [6] や CABIN [18] のような大型のスクリーンを利用した没入型投影技術 (Immersive Projection Technology, 以下 IPT) が注目を集めている。これらのシステムによる視覚提示効果は非常に良好であり、VR の視覚提示装置として様々なアプリケーションを実行するプラットフォームとしての展開が期待される。

しかしながら、これらのディスプレイの現在までの利用形態はコンピュータグラフィックス画像の提示や多面テレビとしてのビデオ再生などが主流であり、操作者の頭部運動による視点移動に実時間で対応した正確な実環境の立体表示、すなわち実空間へのトレイグジスタンスには利用されていない。このような利用形態は IPT に限ったことではなく、一面のスクリーンあるいは CRT を用いたシステムなどの、スクリーンが周囲の空間に対して固定された視覚提示システム (Head-Tracker Display: HTD)[29] 全体に共通の傾向である。実際、操作者に対して視覚的に正確な三次元環境提示を行うトレイグジスタンスシステム (TELESAR [61] [62] [66], TOPS [53] など) に利用されているのは頭部搭載型装置 (Head-Mounted Display: HMD) のみであり、IPT をそのような用途に用いた事例は存在しない。

近年、複数の IPT を結んで実写画像をやりとりし、臨場感通信を行おうという試み [21] が行われるなど、IPT において CG 画像だけでなく実写画像のコンテンツやアプリケーションが注目され始めている。このような背景からも、IPT において実写画像による正確な三次元空間の実時間提示が求められている。現在活発に研究開発が行われている Image Based Rendering 技術 [37] [17] [33] を利用すれば、三次元現実世界を一旦記録した上で自由な視点からの画像を生成して提示することができ、この技術を用いれば記録された世界の中で立体視を行いつつ動き回することは可能である。しかし、この場合も実環境の中を実時間で行動し相互作用を及ぼすことはできない。

本章では、CAVE, CABIN などの IPT やその他の HTD を含むスクリーン固定型視覚提示装置を用いてトレイグジスタンスシステムを構築する場合に、画像の生成系と提示系の間で透視変換の整合性を確保するという観点から解析を行い、操作者の頭部運動に対応した正確な立体画像表示を行うための手法を提案する。また、これに基づくカメラシステムの設計法について述べる。

6.2 固定スクリーンをトレイグジスタンスに利用する際の問題点

6.2.1 形状の変化する非対称透視変換

本章の目的は、上記のような利点を持つスクリーン固定型ディスプレイを頭部運動追従型トレイグジスタンスシステムに適用することにより、操作者に対して負担が少ないシステムを構築することにある。しかし、第 4 章で示したように、固定スクリーンを利用したシステムでは操作者の視点からス

クリーン面へ下した垂線の足は一般にスクリーン中央ではなく、このため非対称透視変換を用いて生成（撮像）した画像を提示する必要がある．こうしたシフト光学系はごく一部の特殊なレンズで実現されているのみであり、実環境の立体提示を行う際に光軸に対して対称な視野を持つ通常のカメラを使用することはできない．さらに、操作者が頭部運動（体幹運動による並進も含む）を行った場合、操作者の視点とスクリーンとの位置関係が変動するので、透視変換のビューイングボリューム形状を「実時間で制御することが必要になる．コンピュータグラフィックス画像を提示する場合は画像を一枚生成する度に透視変換行列を設定すればよいので何ら問題はないが、実写画像をカメラで撮影して実時間で提示する場合には、撮像系の透視変換がカメラの光学系により決定されているので調整が困難である．この問題の解決が IPT を実空間へのトレイグジスタンスに利用するための鍵であると考えられる．

6.2.2 頭部運動追従の必要性

固定スクリーンの利用により比較的安定した画像提示が行えることは既に触れたが、それは頭部運動に追従した画像提示を行わなくてもよいことを意味するものではない．このことを明確にするため、仮に固定した通常のステレオカメラを用いて画像提示を行っている時に操作者が頭部運動を行った場合、どのような影響が生じるかについて解析を行う．通常のステレオカメラをスクリーン方向へ向けて固定し撮影した画像を提示したまま操作者が頭部を回転・並進運動させた場合、提示される物体位置に生じる影響を図 6.1 に示す．初期状態において操作者はスクリーン正面を向いており、図中で左右視点位置はそれぞれ E_R, E_L にあって、その状態で点 P の映像がスクリーン上の点 S_R, S_L に投影されているものとする．点 P は操作者から見てスクリーンの遠方に位置するよう描かれているが、スクリーン上の右目、左目の対応を逆にすることにより手前側の点も全く同様の議論が成り立つ．操作者の眼間距離を d 、操作者の両眼中央 O からスクリーンまでの距離を D と表記しておく．

図 6.1 (a) に示されるように、操作者が両眼の中点を中心に角度 θ だけ頭部を回転させた場合、視点の位置はそれぞれ E'_R, E'_L となる．この状態でスクリーン上の画像が更新されない場合、点 $P(x_p, y_p, z_p)$ は点 $P'(x'_p, y'_p, z'_p)$ の位置に存在するよう見える．以下、点 P' の座標を導出する．

両眼中央 O を原点にとり、 x 軸を右側、 z 軸をスクリーン手前側（図の下側）にとると、点 E_R, E_L の座標はそれぞれ

$$E_R \left(\frac{d}{2}, 0, 0 \right) \quad (6.1)$$

$$E_L \left(-\frac{d}{2}, 0, 0 \right) \quad (6.2)$$

である．点 P を提示するためスクリーン上に投影された左右両眼用の点の座標はそれぞれ

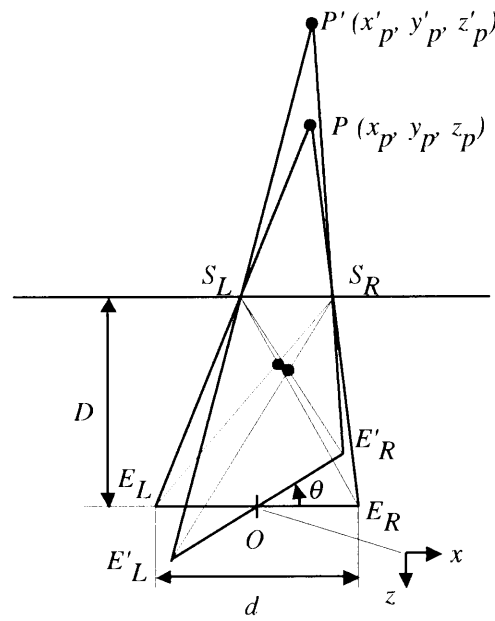
$$S_R \left(\frac{d}{2} - \frac{D}{z_p} \left(x_p - \frac{d}{2} \right), -\frac{D}{z_p} y_p, -D \right) \quad (6.3)$$

$$S_L \left(-\frac{d}{2} - \frac{D}{z_p} \left(x_p + \frac{d}{2} \right), -\frac{D}{z_p} y_p, -D \right) \quad (6.4)$$

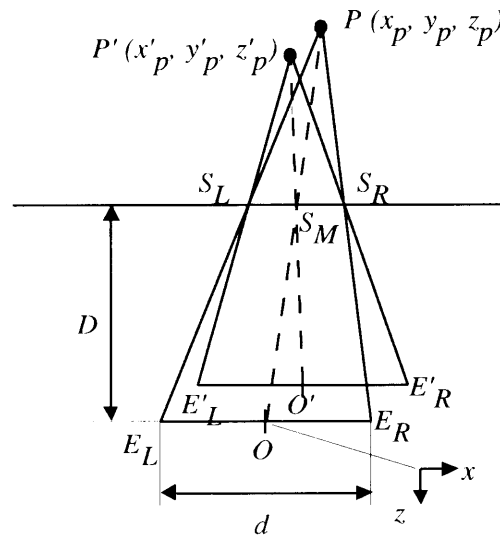
と表される．途中の計算における表記を簡略にするため、点 S_R, S_L の x, z 座標をそれぞれ X_R, X_L と表しておく．すなわち、

$$X_R = \frac{d}{2} - \frac{D}{z_p} \left(x_p - \frac{d}{2} \right) \quad (6.5)$$

$$X_L = -\frac{d}{2} - \frac{D}{z_p} \left(x_p + \frac{d}{2} \right) \quad (6.6)$$



(a) 回転運動による影響



(b) 並進運動による影響

図 6.1 頭部運動に追従しない画像提示を行った場合の影響

また、両眼の中点を中心に頭部が θ 回転した場合の左右両眼の位置は

$$E'_R \left(\frac{d}{2} \cos \theta, 0, -\frac{d}{2} \sin \theta \right) \quad (6.7)$$

$$E'_L \left(-\frac{d}{2} \cos \theta, 0, \frac{d}{2} \sin \theta \right) \quad (6.8)$$

である．

P', S_R, E'_R および P', S_L, E'_L が一直線上にあることから，変数 s, t をパラメータとして

$$\overrightarrow{OE_R} = \overrightarrow{OP'} + s \overrightarrow{P'S_R} \quad (6.9)$$

$$\overrightarrow{OE_L} = \overrightarrow{OP'} + t \overrightarrow{P'S_L} \quad (6.10)$$

と記述される．(6.9) 式における x 座標， z 座標を書き下すとそれぞれ

$$\frac{d}{2} \cos \theta = x + s(X_R - x) \quad (6.11)$$

$$\frac{d}{2} \sin \theta = z + s(-D - z) \quad (6.12)$$

である．(6.12) 式より，

$$s = \frac{\frac{d}{2} \sin \theta + z}{D + z} \quad (6.13)$$

これを (6.11) 式に代入して，

$$-\frac{d}{2} \cos \theta (D + z) + x(D + z) + \left(\frac{d}{2} \sin \theta + z \right) (X_R - x) = 0 \quad (6.14)$$

より

$$\left(D - \frac{d}{2} \sin \theta \right) x + \left(X_R - \frac{d}{2} \cos \theta \right) z + \frac{d}{2} (-D \cos \theta + X_R \sin \theta) = 0 \quad (6.15)$$

を得る．同様に， P', S_L, E'_L が一直線上にあること ((6.10) 式) から

$$\left(D + \frac{d}{2} \sin \theta \right) x + \left(X_L + \frac{d}{2} \cos \theta \right) z + \frac{d}{2} (D \cos \theta - X_L \sin \theta) = 0 \quad (6.16)$$

を得る．(6.15) 式および (6.16) 式を連立させて x, z に関して解くと点 P' の位置が求められる．

z, x に関する項を消去するためそれぞれ (6.15) 式 $\times (X_L + \frac{d}{2} \cos \theta)$ - (6.16) 式 $\times (X_R - \frac{d}{2} \cos \theta)$ および (6.15) 式 $\times (D + \frac{d}{2} \sin \theta)$ - (6.16) 式 $\times (D - \frac{d}{2} \sin \theta)$ を計算すると

$$\begin{aligned} & \left\{ \left(D - \frac{d}{2} \sin \theta \right) \left(X_L + \frac{d}{2} \cos \theta \right) - \left(D + \frac{d}{2} \sin \theta \right) \left(X_R - \frac{d}{2} \cos \theta \right) \right\} x \\ & + \frac{d}{2} \left\{ (-D \cos \theta + X_R \sin \theta) \left(X_L + \frac{d}{2} \cos \theta \right) - (D \cos \theta - X_L \sin \theta) \left(X_R - \frac{d}{2} \cos \theta \right) \right\} = 0 \end{aligned} \quad (6.17)$$

$$\begin{aligned} & \left\{ \left(D + \frac{d}{2} \sin \theta \right) \left(X_R - \frac{d}{2} \cos \theta \right) - \left(D - \frac{d}{2} \sin \theta \right) \left(X_L + \frac{d}{2} \cos \theta \right) \right\} z \\ & + \frac{d}{2} \left\{ \left(D + \frac{d}{2} \sin \theta \right) (-D \cos \theta + X_R \sin \theta) - \left(D - \frac{d}{2} \sin \theta \right) (D \cos \theta - X_L \sin \theta) \right\} = 0 \end{aligned} \quad (6.18)$$

整理して，

$$\begin{aligned} & \left\{ -D(X_R - X_L) + Dd \cos \theta - \frac{d}{2} \sin \theta (X_R + X_L) \right\} x \\ & + \frac{d}{2} \left\{ -D \cos \theta + (X_R + X_L) + 2X_R X_L \sin \theta + \frac{d}{2} \cos \theta \sin \theta (X_R - X_L) \right\} = 0 \end{aligned} \quad (6.19)$$

$$\begin{aligned} & \left\{ D(X_R - X_L) - Dd \cos \theta + \frac{d}{2} \sin \theta (X_R + X_L) \right\} z \\ & + \frac{d}{2} \left\{ -2D^2 \cos \theta + D \sin \theta (X_R + X_L) + \frac{d}{2} \sin^2 \theta (X_R - X_L) \right\} = 0 \end{aligned} \quad (6.20)$$

ここで, (6.5) および (6.6) 式より

$$X_R + X_L = -\frac{2Dx_p}{z_p} \quad (6.21)$$

$$X_R - X_L = d + \frac{Dd}{z_p} = d \left(1 + \frac{D}{z_p} \right) \quad (6.22)$$

$$X_R X_L = \left(\frac{D}{z_p} x_p \right)^2 - \frac{d^2}{4} \left(1 + \frac{D}{z_p} \right)^2 \quad (6.23)$$

であるから, これらを代入し,

$$\begin{aligned} & \left\{ -Dd \left(1 + \frac{D}{z_p} \right) + Dd \cos \theta + Dd \frac{x_p}{z_p} \sin \theta \right\} x \\ & + \frac{d}{2} \left\{ 2D^2 \cos \theta \frac{x_p}{z_p} + 2 \left(\left(D \frac{x_p}{z_p} \right)^2 - \frac{d^2}{4} \left(1 + \frac{D}{z_p} \right)^2 \right) \sin \theta + \frac{d^2}{2} \sin \theta \cos \theta \left(1 + \frac{D}{z_p} \right) \right\} = 0 \end{aligned} \quad (6.24)$$

$$\begin{aligned} & \left\{ Dd \left(1 + \frac{D}{z_p} \right) - Dd \cos \theta - Dd \frac{x_p}{z_p} \sin \theta \right\} z \\ & + \frac{d}{2} \left\{ -2D^2 \cos \theta - 2d^2 \frac{x_p}{z_p} \sin \theta + \frac{d^2}{2} \sin^2 \theta \left(1 + \frac{D}{z_p} \right) \right\} = 0 \end{aligned} \quad (6.25)$$

さらに整理して

$$\begin{aligned} & \left\{ \left(1 + \frac{D}{z_p} \right) - \cos \theta - \frac{x_p}{z_p} \sin \theta \right\} x \\ & = D \left\{ \frac{x_p}{z_p} \cos \theta + \left(\left(\frac{x_p}{z_p} \right)^2 - \left(\frac{d}{2D} \right)^2 \left(1 + \frac{D}{z_p} \right)^2 \right) \sin \theta + \left(\frac{d}{2D} \right)^2 \left(1 + \frac{D}{z_p} \right) \sin \theta \cos \theta \right\} \end{aligned} \quad (6.26)$$

$$\left\{ 1 + \frac{D}{z_p} - \cos \theta - \frac{x_p}{z_p} \sin \theta \right\} z = D \left\{ \cos \theta + \frac{x_p}{z_p} \sin \theta - \left(\frac{d \sin \theta}{2D} \right)^2 \left(1 + \frac{d}{z_p} \right) \right\} \quad (6.27)$$

表記を簡単にするため, 式の中で頻繁に出現する項を

$$\alpha = x_p / z_p \quad (6.28)$$

$$\beta = 1 + \frac{D}{z_p} \quad (6.29)$$

$$\gamma = \frac{d}{2D} \quad (6.30)$$

のように表記すると, 最終的に点 P' の x, z 座標は

$$x'_p = D \frac{\alpha \cos \theta + (\alpha^2 - \gamma^2 \beta^2) \sin \theta + \gamma^2 \beta \sin \theta \cos \theta}{\beta - (\cos \theta + \alpha \sin \theta)} \quad (6.31)$$

$$z'_p = D \frac{\cos \theta + \alpha \sin \theta - \gamma^2 \beta \sin^2 \theta}{\beta - (\cos \theta + \alpha \sin \theta)} \quad (6.32)$$

と求められる.

計算された空間歪みの様子を図 6.2 に示す. ここで用いているパラメータ値は, 眼間距離 d が 6.5 [cm], 頭部回転角 θ が 10 [deg], 両眼の中央からスクリーンまでの距離 D が 2.0 [m] である. 計算に

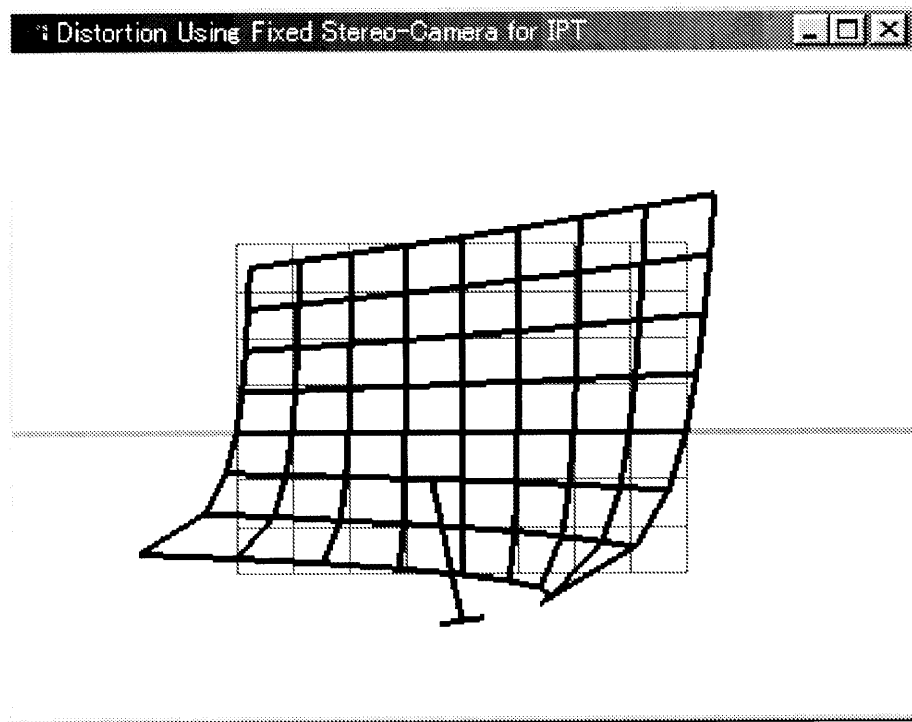


図 6.2 頭部運動に追従しない画像提示時に頭部回転運動を行った場合の空間歪み

用いた格子点は奥行き方向が 0.5 [m] から 4 [m] まで，横方向が左右 2 [m] ずつの範囲内で，ともに 0.5 [m] 刻みである．スクリーン近傍を除いて，空間全体が大きく歪んだ形で操作者に提示される結果となることが示される．

この歪みをもう少し詳しく解析してみる．(6.31), (6.32) 式において， α は注目する点の方向， β は注目する点の奥行きに関するパラメータ， γ はスクリーンまでの距離に対する眼間距離の比の半分である．人間を包囲する大きさのスクリーンを使用する場合，スクリーンまでの距離のオーダーはメートル，眼間距離のオーダーはセンチメートルであるため， γ の値のオーダーは 1/100 である．(6.31) および (6.32) 式において， γ は 2 乗の形で入ってくるため，これらは他の項と比べて非常に小さい．このため，

$$x'_p \approx D \frac{\alpha \cos \theta + \alpha^2 \sin \theta}{\beta - (\cos \theta + \alpha \sin \theta)} \quad (6.33)$$

$$z'_p \approx D \frac{\cos \theta + \alpha \sin \theta}{\beta - (\cos \theta + \alpha \sin \theta)} \quad (6.34)$$

であり，このとき

$$\frac{x'_p}{z'_p} \approx \alpha = \frac{x_p}{z_p} \quad (6.35)$$

となる．この結果は，空間中の点の方向が元の状態と比べてほとんど変化しないことを示している．言い換えれば，固定スクリーン，特に大画面を利用する場合に固定ステレオカメラを使用するシステムにおいて操作者が頭部回転運動を行った場合，世界は横方向には揺れず，ほぼ奥行き方向のみに歪むことが示される．この様子は，図 6.3 のようにメッシュを同心円状に切ると観察しやすい．

次に，操作者頭部の平行移動による影響を図 6.1 (b) に示す．こちらはより単純であり，過去にもステレオ画像提示の研究の中でその影響が計算されている [7] [51] が，ここで整理しておく．操作者

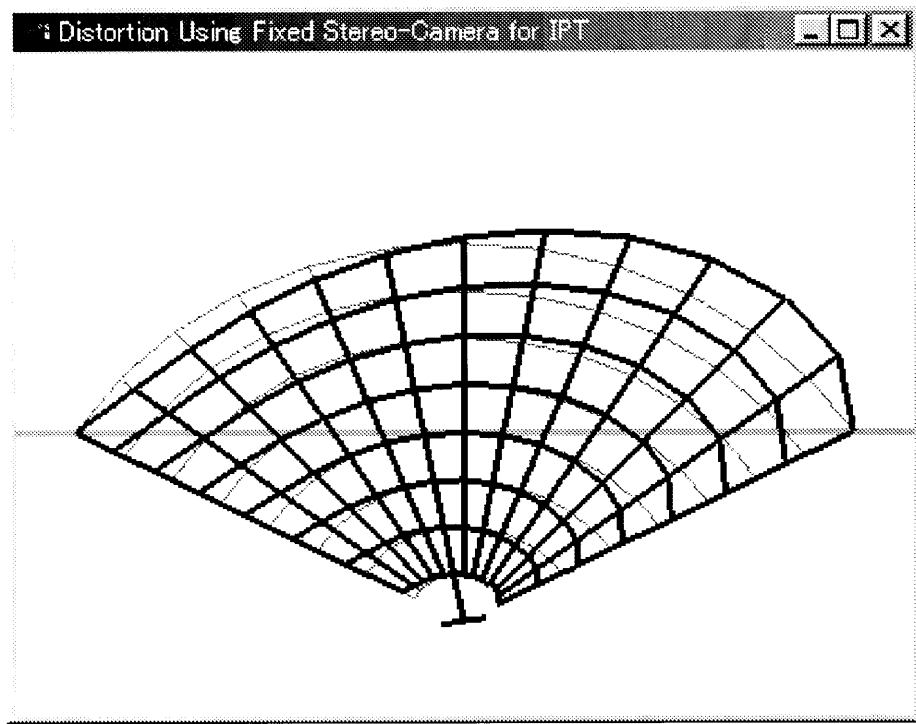


図 6.3 頭部運動に追従しない画像提示時に頭部回転運動を行った場合の空間歪み（扇形メッシュ）

の両眼中点位置が原点 O からへ平行移動した場合、画像上の点は P から P' へ動いて観察される．スクリーン上に投影された点 S_R, S_L の中点を S_M とすると、

$$\overrightarrow{OS_M} = -\frac{D}{z_p}\overrightarrow{OP} \quad (6.36)$$

である．直線 $OP, O'P'$ はともに点 S_M を通るので

$$\begin{aligned} \overrightarrow{OP'} &= \overrightarrow{OO'} + \overrightarrow{O'P'} \\ &= \overrightarrow{OO'} - \frac{z_p}{D}\overrightarrow{O'S_M} \\ &= \overrightarrow{OO'} - \frac{z_p}{D}(\overrightarrow{OS_M} - \overrightarrow{OO'}) \\ &= \overrightarrow{OP} + \left(1 + \frac{z_p}{D}\right)\overrightarrow{OO'} \end{aligned} \quad (6.37)$$

となる．これは、点 P がスクリーンより手前 ($z > -D$) ならば操作者と同じ方向へ、遠方 ($z < -D$) ならば操作者と逆方向へ、点 S_M を中心として空間全体が圧縮・伸長しながら動くことを示している．スクリーンまでの距離と同じ位置にあるすべての点は頭部運動による影響を受けず、このような点およびスクリーン近傍の点は頭部運動による影響が小さくなっている．

人間が自然に頭部を動かした場合、視点の移動は両眼中点を中心とする回転成分と、首からのオフセットに伴う平行移動成分とを合成したものとなる．このため、操作者の頭部運動により世界全体が歪むことが示される．

6.2.3 カメラの光軸方向の問題

ビューイングボリューム形状制御の他に、もう一つの問題は、頭部回転運動を行ったときに前額面とスクリーンとの姿勢が変化し平行でなくなってしまうことである．提示面と平行でない撮像面に投

影された画像は、二次元的な操作だけでは本来の画像を復元できず、バーチャルなスクリーンを想定してテキストチャマッピングなどにより三次元的な補正を行わない限りは正確な輻輳・視差情報を操作者に提示することができない。三次元グラフィックスハードウェアの性能向上が著しい現在ではテキストチャマッピングを普及価格帯のパーソナルコンピュータ上でほぼ実時間処理することが可能になっており、この要求はもはやコスト的にそれほど高くつくものではなくなっている。しかしながら、この処理を行うと画像取得から提示までの処理過程が一段階増えることになるので、頭部運動に追従した実時間提示という観点からは歓迎されることではない。さらに、HMD 利用システムで行われているようなロボット頭部に固定したカメラを用いると、光軸方向がスクリーンの端を向いてしまい、有効な画像提示領域が著しく小さくなってしまうという問題が残る。

従来、固定スクリーンを用いて実写画像の立体提示を行う場合は、あらかじめカメラの光軸をある注視点へ向けておき、その注視点近傍の視差をディスプレイ上に表示するという手法がよく用いられてきた。こうした立体画像撮影・提示方式はそもそも撮像系と提示系とでビューイングボリュームの形状や位置関係が異なっているため、輻輳と視差を手がかりとした正確な空間提示は行えない。しかしながら、こうした方式に従って立体画像を提示した場合の距離認識誤差に関する解析 [7] [28] などが行われており、提示スクリーンに対して傾いた撮像面により得られた画像を無造作にスクリーンへ表示することにより生じる悪影響を容易に推察することができる。

6.3 テレイグジスタンス視覚系の構築手法

前節の議論より、スクリーン固定型視覚提示システムを用いてテレイグジスタンスシステムを構成する場合の課題は、以下のようにまとめられる。

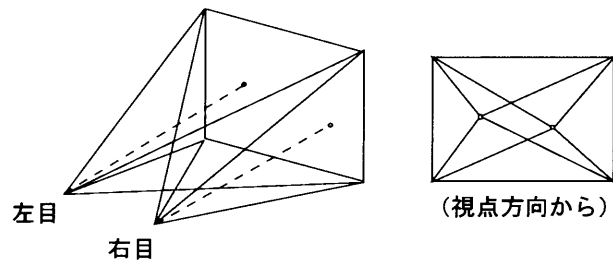
1. 操作者の頭部運動をトラッキングして、視点とスクリーンの位置関係に対応したビューイングボリューム形状制御を行うこと。
2. カメラの撮像面が常にスクリーンと平行になるよう制御すること。この課題は、単純な二次元的画像処理（位置調整および拡大縮小）のみでシステムを構築可能とし、スクリーンの有効表示領域をなるべく損なわないことから来る要請である。

これらの課題に対して、操作者の視点とスクリーンとの位置関係変化に追従して実時間で適切な対応をとることができれば、スクリーン固定型提示装置を用いたテレイグジスタンス視覚系が実現できる。

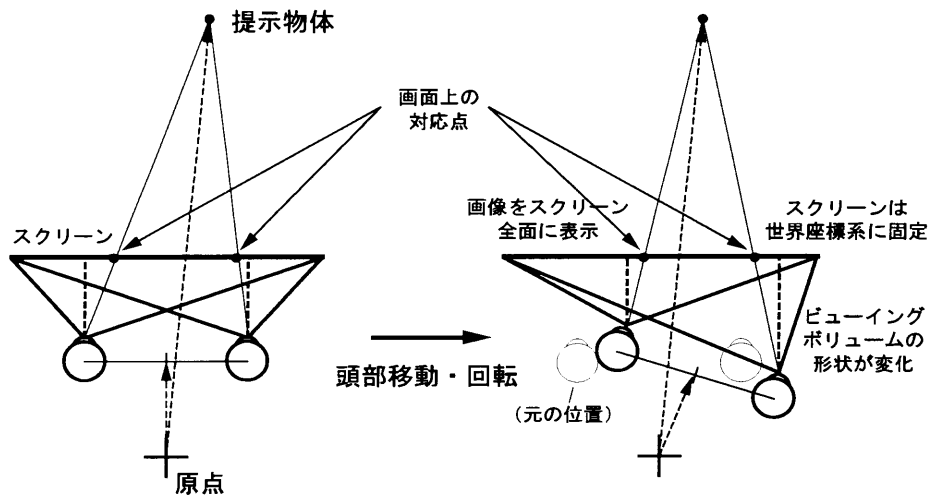
上記のタスクをシステムへ実装するにあたって、撮像系と提示系のどちらにどの役割を分担させるかによっていくつかの選択肢が存在する。なお、ここではカメラにより2次元画像が生成されるまでを「撮像系」、それ以降最終的に操作者へ画像が提示されるまですべてを「提示系」と呼ぶ。最初に、コンピュータグラフィックス画像提示の場合と同様に、撮像（画像生成）系においてすべて対応する方法が考えられる。この場合、システムは以下の構成で実現される。

Type A: 撮像系における透視変換のビューイングボリューム形状（非対称四角錐）を実時間で制御するために、光軸方向とズーム倍率を実時間で制御可能な特殊なカメラを用いる。画像提示系はCG画像を提示する手法 (図 6.4) と全く同一である。

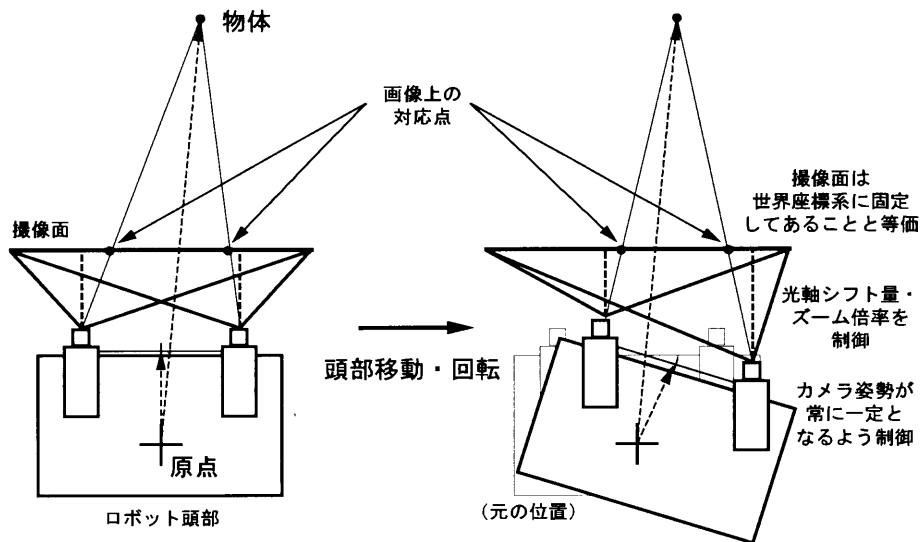
このように特殊なカメラを設計することも問題解決の一つの方向であるが、本研究ではさらに光軸が撮像面に対して垂直で、光軸に関して上下左右対称な視野を持つ通常のカメラを利用してテレイグジスタンスシステムを実現する方法を提案する。これはすなわち、ビューイングボリューム形状制御と等価な作業の全体または一部を画像提示系において行うことにより実現するというアプローチである。



(a) ビューイングボリュームの形状



(b) 提示系における頭部運動への対応



(c) 撮像系における頭部運動への対応

図 6.4 スクリーン固定型システムの透視変換 (Type A: 非対称可変四角錐, CG 用/撮像側で対応)

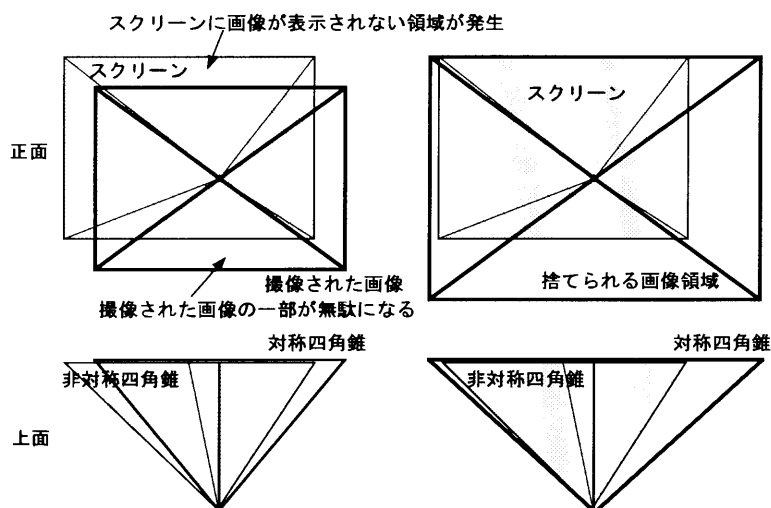


図 6.5 非対称四角錐ビューイングボリュームの対称四角錐による代替

前述の課題1において、カメラにより撮影された画像全体が無駄なくスクリーン上に投影され、かつスクリーンの全領域に常に画像が表示されるためには、視点とスクリーンにより構成される非対称四角錐の形状を撮像系により実現する必要がある。しかしながら、カメラで撮影された画像の端が無駄になる、あるいは必ずしもスクリーン全体に画像が表示されていなくてもよいという状況を許せば、この問題は撮像系における視野角（ズーム倍率）制御か、あるいは提示系におけるスクリーン上画像の表示位置制御および拡大縮小という問題に帰着する。

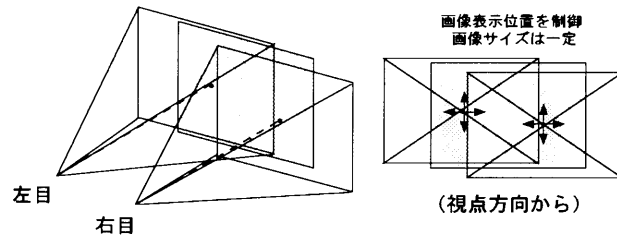
非対称四角錐形状のビューイングボリュームにより得られる画像を、対称四角錐形状のビューイングボリュームを用いて等価的に実現する方法を図6.5に示す。図6.5上半分は視点からスクリーン方向を見たもの、下半分はスクリーン上方から見た図である。視点からスクリーン面へ下した垂線の足が提示される画像の中央となるように、対称四角錐形状のビューイングボリュームにより生成（撮像）された画像を提示する。この際、非対称四角錐の場合と対称四角錐の場合とでスクリーン上に表示される画像が一致するようにしておけば、画像が表示されている部分に関しては全く等価である。図6.5左に示すように一般的には画像が表示されないスクリーン上の領域が生じたり、撮像された画像が有効に利用されないなどの損失が生じる。ここで図6.5右に示すように撮像系として広角のカメラを用いると、有効に利用されない画像領域の割合が大きくなるため画像解像度の面では不利であるが、操作者のある程度の動きに対してスクリーン全体をカバーしスクリーン上全ての領域に画像を表示することは可能である。

なお、この操作において、生じる損失はこうしたスクリーンや画像の領域の問題のみであり、このことにより新たな画像歪みが生じるものではない。

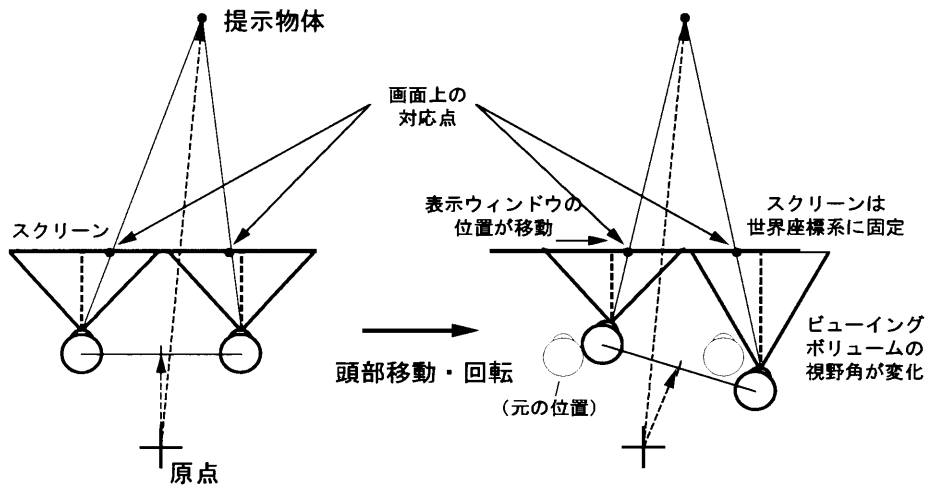
光軸が撮像面と垂直な通常のカメラを用いる場合、撮像系と提示系の役割分担に応じて以下の2通りの方法が考えられる。

Type B: 画像提示系において、画像中央が視点からスクリーンへ下した垂線の足に一致するよう画像表示位置を制御し、表示画像サイズはスクリーン上で一定とする。この場合、頭部運動により視点のスクリーンからの距離が変化し画像を見込む視野角が変化するので、カメラのズーム倍率を実時間で制御する(図6.6)。

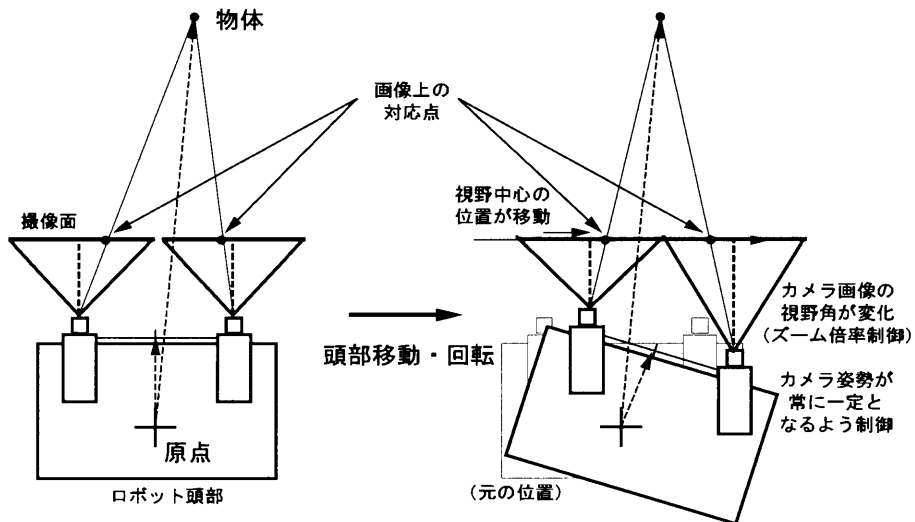
Type C: 通常のカメラを用い、ズーム倍率も一定とする。すなわち、ビューイングボリューム形状は常に一定である。スクリーンと視点の位置関係変化を吸収するため、画像提示系においてType B



(a) ビューイングボリュームの形状

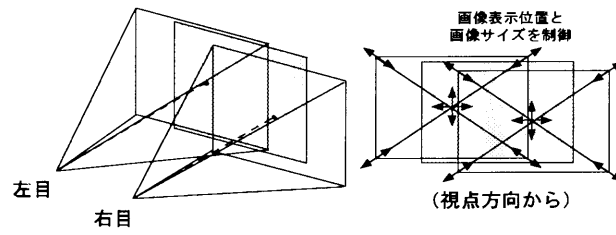


(b) 提示系における頭部運動への対応

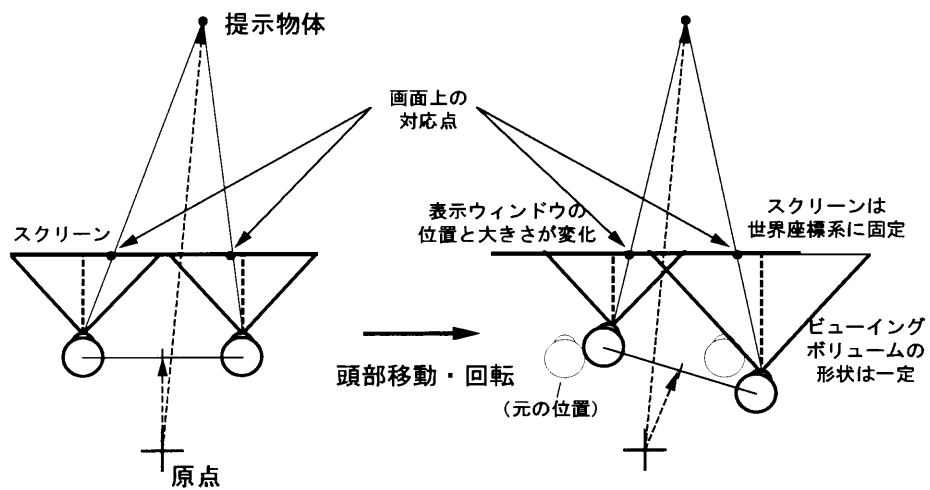


(c) 撮像系における頭部運動への対応

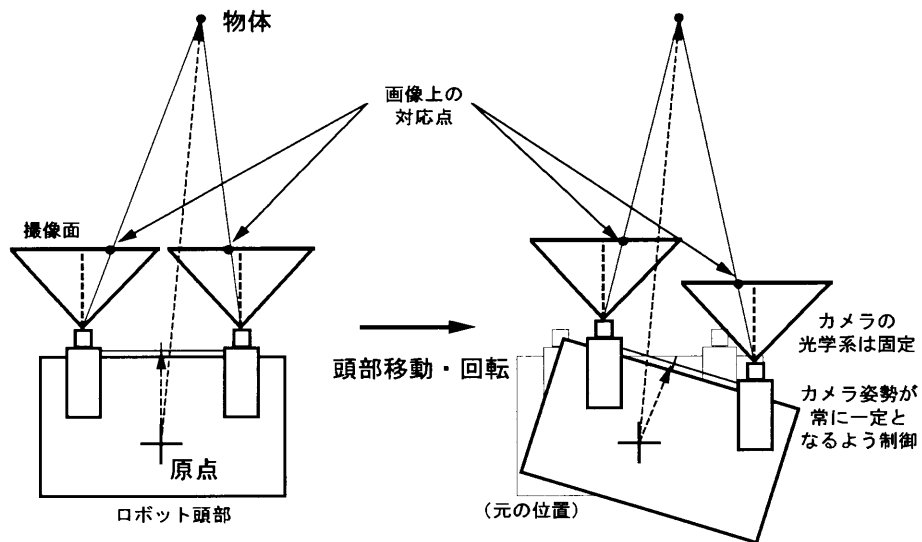
図 6.6: スクリーン固定型システムの透視変換と撮像・提示系の役割 (Type B: 対称視野角可変四角錐)



(a) ビューイングボリュームの形状



(b) 提示系における頭部運動への対応



(c) 撮像系における頭部運動への対応

図 6.7: スクリーン固定型システムの透視変換と撮像・提示系の役割 (Type C: 対称視野角固定四角錐)

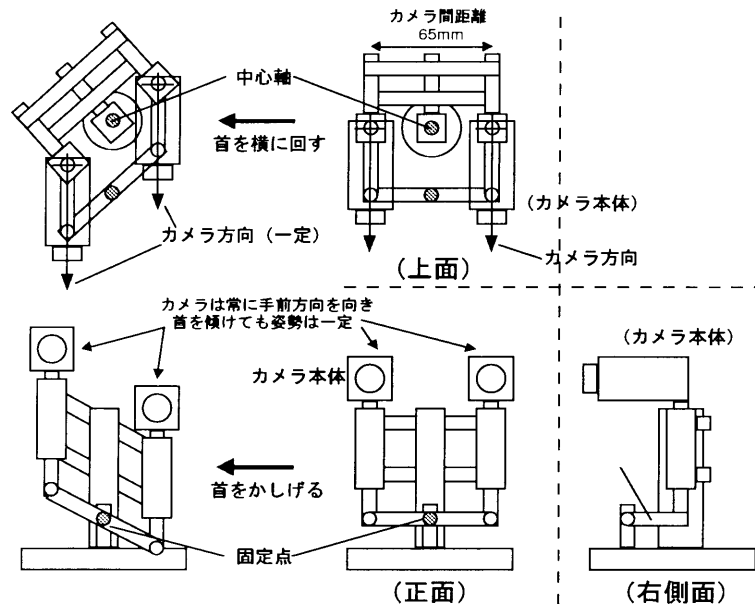


図 6.8 姿勢保持機構のコンセプト

と同様に画像表示位置を制御するとともに、スクリーン上の表示画像サイズを制御して視野角を一定に保つ(図 6.7)。

Type B の場合、画像提示系の処理は画像提示位置のシフトだけなので単純である。その反面、画像生成側でカメラのズーム倍率を操作者の頭部運動にあわせて実時間で制御することが必要であり、カメラのズーム倍率制御性能が所望のレベルを達成できるかどうかの実現の鍵となる。Type C の場合、カメラの光学系は固定であり、撮像系の構築が最も簡単となる。この場合、画像の拡大縮小操作が実時間で遅れを伴うことなく行われることが要求される。なお、提示系における画像の位置制御や拡大縮小は、電子的な操作および提示デバイスレベルの電氣的/光学的操作などいくつかの選択肢が考えられる。市販のプロジェクタなどを利用する提示システムの場合、デバイスレベルでの操作は困難であり、電子的な画像処理装置を利用の方が妥当である。汎用的な PC もこのような画像処理装置として利用することが十分可能である。

以上、撮像系と提示系の役割分担に応じて 3 種類の方法を示した。

6.4 カメラシステムの設計

6.4.1 姿勢保持機構の設計

前節で述べた 3 種類の方法でシステムを構築する際に共通して必要なのは、図 6.4 (c)、図 6.6 (c)、図 6.7 (c) のように、操作者の頭部運動に関わらずカメラの姿勢を常に一定に保つことである。すなわち、操作者サイトの視点とスクリーンの位置関係を遠隔サイトにコピーした状況を想定して、スクリーンとカメラの撮像面が常に平行となるようにする。光軸が撮像面と垂直な通常のカメラを利用する場合は、光軸が常にスクリーンと垂直な関係を保つようにする。

この機能を実現するにあたって、カメラの姿勢を頭部に対して独立に制御する方法も考えられるが、姿勢を一定に保つという目的のためには、いたずらに制御自由度を増やさず、カメラが体幹に対して常に同じ姿勢をとるようなリンク機構を利用することが得策であると考えられる。

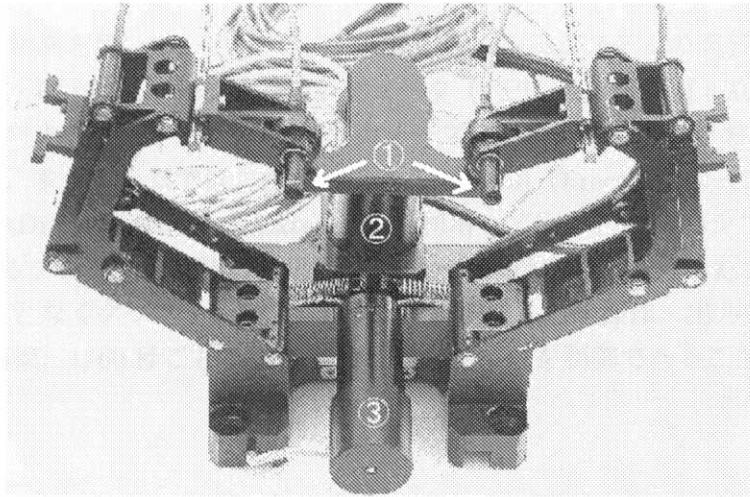


図 6.9 試作した姿勢保持リンク機構：① カメラ，② yaw 軸モータ，③ roll 軸モータ

図 6.8 に姿勢保持機構のコンセプトを示す．必要な機能は，首を傾ける動作 (rolling) と左右回転動作 (yawing) の 2 自由度に対してカメラの姿勢を一定に保つことである．頷きのような pitching 動作に関しては，左右の視点の位置変化が等しくなるので，機構全体を平行移動させればよい．最終的に，本機構全体を 3 自由度平行移動可能なステージの上に載せて頭部運動に追従させれば，スクリーン固定型視覚提示装置を用いて頭部運動に対応したトレイグジスタンス視覚システムを構築できる．

このコンセプトモデルに基づいて，姿勢保持機構の設計・試作を行った．試作した機構を図 6.9 に示す．機構は，カメラ位置制御用の 2 自由度シリアルリンクと，姿勢保持用の平行リンクおよびスライド機構の組み合わせで構成されている．本機構は，人間の首の運動が yaw 方向の回転に対して高速である特性を考慮し，yaw 方向の慣性モーメントが小さくなるよう設計した．シリアルリンクは roll 方向回転軸の先に yaw 方向回転軸を備えており，両側の平行リンク機構とその先の前後方向スライド機構により，シリアルリンクの先に取り付けられたカメラの姿勢を一定に保つようになっている．また，本機構だけで人間の首の運動にある程度対応できるように，シリアルリンクにオフセットを持たせることにより，純粋な回転自由度だけでなく人間の頭部回転に伴い発生する自然な並進運動が生じるようになっている．リンクのオフセットは上下方向 125 [mm]，前後方向 50 [mm] であり，普通の成人のサイズと比べてやや小さいが，オフセット値を大きくすると慣性モーメントの増加が生じるので，運動性能とのバランスを考えてこのようなサイズとした．

制御に必要な性能を決定するにあたり，人間の yaw 方向の頭部回転運動に関しては最高速で動かした場合 120 [deg] の範囲を 0.4[s] で往復することを実測により概算し，これに単純な等加速度加減速プロファイルを当てはめて，最大角加速度を 1.2×10^4 [deg/s²] と算出した．可動範囲は yaw 方向 70 [deg]，roll 方向 30 [deg] とする．カメラ耐荷重は，接続ケーブルなどの影響を考慮して 200 [g] としている．

本機構の 2 自由度の関節を DC モータ (Maxon 社製，roll 方向 RE025-055: 20 [W]，yaw 方向 RE036-072: 70 [W]) により駆動し，ロータリーエンコーダ (多摩川精機製 OIH35, 3000 [ppr]，通倍なし) により角度を読み出す．姿勢保持機構の制御は，汎用のパーソナルコンピュータを用いて行っている．搭載カメラは直径 7 [mm] の小型カメラ (東芝製 IK-SM43H: 1/4 インチ 41 万画素) および同カメラ用レンズ (東芝製 JK-L04S: 焦点距離 $f = 4$ [mm]，水平視野角 51 [deg]，垂直視野角 39 [deg]) を使用している．

6.4.2 提示用画像操作部の構築

本姿勢保持リンク機構を用いて、Type C 方式による画像提示システムの実装を行った。画像の二次元的なシフトと拡大・縮小は、パーソナルコンピュータ（CPU Pentium-II 450MHz, メモリ 128MB）用のビデオ出力機能付きグラフィックスボード（カノーブス社製 Spectra2500 および同社製 VideoPort 600）により行った。本システムでは、画像入力を Microsoft DirectX（バージョン 5 以降）の機能の一部である DirectDraw Video Port Extension を利用して行い、画像のシフトおよび拡大縮小を DirectDraw におけるブロック転送時の転送先バッファ領域指定により行った。この構成は、システムバスを経由することなくグラフィックスボード上で画像の入力・操作・出力を行うことを意図したものである。この結果、1/60 秒ごとに画像表示位置とサイズを制御できることを確認した。

6.5 まとめ

IPT ディスプレイを含むスクリーン固定型視覚提示装置を使って頭部運動に追従した実写画像による正確な三次元空間提示を実時間で行おうとした場合、操作者の頭部運動に従ってビューイングボリュームの形状が変化することと、カメラ撮像面とスクリーンとの平行が保てないことに問題があることを明らかにした。この解析に基づいて、スクリーン固定型装置を用いて三次元実空間を正確に提示するための原理を示した。すなわち、操作者が頭部運動を行っても視点間距離を一定に保ったままカメラの光軸をスクリーンと垂直な姿勢に保持する仕組みを導入し、かつビューイングボリュームの形状を制御するという方法である。ここで行われるビューイングボリューム形状制御は、非対称四角錐の形状を撮像系で制御するような光学系を実現する方法以外に、対称四角錐形状のビューイングボリュームを用いて生成された画像を用い、提示画像の位置とサイズを制御することにより等価的に実現可能であることを示した。最後に、このようなシステムを構築する際に必要な技術要素である、カメラ姿勢保持機構の設計と試作、および画像の実時間操作を行うサブシステムの構築を行い、本手法が工学的に実現可能であることを示した。

本章で提示した手法は、固定スクリーン型視覚提示システムにおいて、頭部運動に対応したコンピュータグラフィックス画像の立体提示と同等の機能を実写画像に対して実現するものであり、理想的な光学系の下では近似のない正確な輻輳と視差の手がかりを操作者に提示できる。その意味で、CAVE における頭部回転運動に対する良好な特性は全て本手法を適用したトレイグジスタンスシステムに受け継がれる。

以上の結果により、操作者の頭部回転運動に対して安定した画像提示が可能なスクリーン固定型画像提示システムを用いてトレイグジスタンスを行うための筋道が開かれたと考える。本手法は固定スクリーンを利用する視覚提示システムに対して汎用的な技術であり、IPT のような大規模な装置以外にも、通常の HTD システム、例えば一面スクリーンシステム、CRT や液晶ディスプレイのように比較的簡易なトレイグジスタンス視覚システムにも適用可能である。本手法の適用用途としては、操作者の周囲全体の見回し動作を必要としないが高精度で安定した画像提示が要求されるような用途、例えば遠隔手術システムなどへの適用が考えられる。

第 7 章

結論

本論文では、バーチャルリアリティおよびトレイグジスタンスにおいて人間が運動を行った場合に提示される世界がどのように観察されるかという視点に立った現象の解析と、安定な空間提示を実現するためのシステム構築に関して記述した。

第 1 章ではバーチャルリアリティおよびトレイグジスタンスシステムの一般的事項について概説し、システムとして体験した場合の「違和感」という漠然とした感覚を、リアルタイムインタラクションの実現、より具体的には操作者の運動と例示される視覚情報との整合性という切り口から見ることで、工学的な解析の俎上に乗せた。

第 2 章では、最も典型的な VR システムの構成、すなわち操作者の運動計測と HMD による VR 空間の提示という構成をとりつつ、極力実時間性を重視したシステムの構築を行った。ハードウェア的には、運動計測装置として物理世界のロボットを制御するためのトレイグジスタンスマスタ装置を利用し、大きな自由度数の運動情報を高速・安定に VR 空間の中へ取り込むことに成功した。ソフトウェア的には、システム全体の仕事をいくつかのサブタスクに分割し仕事の分散を図ると共に、最も負荷が大きく時間がかかる画像生成についてはいくつかの高速化テクニックを開発・実装し、効率の良い VR 空間画像の生成を実現した。これにより、システム全体としての遅延時間を小さくすることができ、運動感覚と提示される視覚との整合性の確保を実現した。同時に、使用する HMD の仕様に対応するパラメータを用いることにより自然な三次元空間の提示を、また大きな自由度を反映して操作者の上肢や体幹の運動を VR 空間中に反映させることにより自己投射性を実現した。

第 3 章では第 2 章で構成したシステムを利用して、空間的な側面と時間的な側面が互いに干渉することを示した。すなわち、両眼立体視を利用する視覚提示デバイスを用いて空間提示を行っている場合に、何らかの理由により正確な空間提示が行えない状況において頭部運動を行うと、空間の動的な変形と不自然な画像のフローが生じるという現象を幾何学的解析により定量的に説明した。また、この現象によって作業性に影響が生じることを実験により確認した。従来 VR システムの時間的な要素といえば純粋な時間遅ればかりが議論されてきたが、ここで指摘した問題は、たとえシステム全体の時間遅れが皆無という理想的な状態でも不正確な立体視を行っている和不具合が生じるということを明らかにしたものである。

ここまでは HMD を用いた VR・トレイグジスタンスシステムの構成に関して話を進めてきたが、第 4 章では、近年各地に建設されている IPT に着目し、HMD を用いたシステムとの比較を行うことにより、それぞれのディスプレイにおける利点と欠点を解析した。その結果、IPT の視覚提示品質が良好であることの本質は頭部搭載でなく空間に固定した画像提示面を使用していることであることを明らかにした。頭部運動、特に回転運動を行った場合における提示画像の安定性は固定スクリーン型の方が優れており、simulator sickness を喚起しにくい性質を持っていることを示した。逆に、固定スクリーンを用いて自然な立体視を行うには、操作者の運動に従って時間的に変動する非対称透視投影によって生成される画像をスクリーン上に表示することが必要であり、HMD を利用した実環境へ

のトレイグジスタンスシステムとは異なり光軸に対称な視野を持つ通常のカメラをそのまま使用できないという問題を指摘し、続く2章へ向けた問題提起を行った。

第5章では、システム全体にわたる時間遅れやCG画像生成時のフレームレートが十分に確保できない場合に、HMDを用いたシステムでは世界が揺れるという弱点の解決を図った。操作者に近い場所にローカルな高速ループを構成することにより、等価的に高フレームレートと遅延の補償を行う手法を提案した。これは視野角が限定されることを除いて、頭部運動に対する画像の振る舞いにおいて等価的に固定スクリーン型システム的良好な性質をHMDにおいて実現するものであり、世界を安定化させる効果を持つ。

第6章では、もともと頭部運動感覚と視覚との整合性確保が容易な固定スクリーン型システムを実環境へのトレイグジスタンスに適用するにあたり、第4章で指摘した、時変非対称透視投影の問題に対する解決法を示した。その際、複雑な三次元空間の再構築や画像の三次元的な変形操作を行わず、あくまでカメラで取得した二次元画像を二次元画像のままで扱うため、カメラの姿勢を常に一定に保つリンク機構を発案・設計・試作して、これと二次元的な画像のシフト・拡大縮小により固定スクリーンを利用した正確な実環境へのトレイグジスタンスが実現できることを示した。

以上のように、本研究ではVRおよびトレイグジスタンスにおいて人間が運動を行った場合に提示される世界がどのように見えるかに関して探求してきた。従来、ともすれば個々のデバイスレベルでの議論に終始し、システムを構築して人間が運動を行った時に生じる現象については見過ごされがちであったが、本研究ではそこに運動感覚と視覚との整合性という観点を導入することにより、インタラクティブなシステムにおいて安定な世界提示を実現するための設計指針を提示した。これは、いわば「止まっている世界が止まって見える」という、現実世界では至極当たり前のことをVRシステムで実現するための非常に基本的かつ重要なポイントであり、VRやトレイグジスタンスが一般社会に浸透していくために必要なマイルストーンを示すことができたと考える。

謝辞

本論文を執筆するにあたり、数多くの人々にご指導、助言や励ましを賜りました。

東京大学大学院工学系研究科 舘 暲 教授には、研究のすべてにわたってご指導と励ましをいただきました。同時に、筆者がこの分野における研究を始める機会を与えて下さり、バーチャルリアリティ研究の先頭に立って数々の研究を推進することにより本研究のバックグラウンドを提供して下さいました。

東京大学大学院工学系研究科 安藤 繁 教授，嵯峨山 茂樹 教授，東京大学国際・産学共同研究センター 満洲 邦彦 教授，東京大学情報学環 前田 太郎 講師には、副査をしていただき、貴重なアドバイスをいただきました。同時に、満洲 教授には、バーチャルリアリティ技術の医療応用へ向けた観点からアドバイスをしていただき、特に第 6 章の研究においてご支援をいただきました。前田 講師は、筆者にとって最も身近に相談できる研究者であり、本研究分野においては鋭い洞察と豊かな発想力により、筆者に多大な影響を及ぼすとともに、数々のヒントと有益なアドバイスをいただきました。

東京大学大学院工学系研究科 石川 正俊 教授には、感覚統合の重要性と実時間システムに関するフィロソフィーを教えていただきました。

東京大学先端科学技術研究センター 廣瀬 通孝 教授 および 通信・放送機構 小木 哲朗 研究員は、東京大学 IML において CABIN を構築し、筆者が IPT による安定な画像提示を体験する機会を作して下さいました。この体験がなければ、第 5 章および第 6 章における研究の発想は困難だったと思われます。

奈良先端科学技術大学院大学 竹村 治雄 助教授 には、VR のソフトウェアに関するアドバイスをいただきました。

東京大学大学院工学系研究科 川上 直樹 助手 並びに東京大学国際・産学共同研究センター 稲見 昌彦 リサーチアソシエイトは、日頃から研究の細部にわたってディスカッションをしていただき、本研究におけるアイデアのヒントをいただきました。

(財) イメージ情報科学研究所 関東研究所の方々には、日頃からバーチャルリアリティの産業応用などに関してディスカッションしていただくとともに、特に第 2 章のソフトウェアシステムの開発と第 5 章の研究においてご支援をいただきました。

また、東京大学大学院工学系研究科 舘研究室の学生諸氏並びに卒業・修了生の方々には、公私にわたり研究生生活を共にする仲間としておつきあいいただきました。

最後に、毎日の生活の中でいつも励まし続けてくれた 妻 みさと、存在そのものが力づけである子供たち、康明と未来 に、すべてを感謝します。

参考文献

- [1] D. L. Akin, M. L. Minsky et al.: “Space Application of Automation: Robotics and Machine Intelligence Systems (ARAMIS) —Phase II,” *NASA Contract Report*, No. 3734, 1983.
- [2] Ronald Azuma: “A Survey of Augmented Reality,” *Presence*, Vol. 6, No. 4, pp. 335-385, 1997.
- [3] Gary Bishop, Henry Fuchs, Leonard McMillan, and Ellen J. Scher Zagier: “Frameless Rendering: Double Buffering Considered Harmful,” *Proceedings of SIGGRAPH '94*, pp. 175-176, Orlando, Florida, 1994.
- [4] John J. Craig: *Introduction To Robotics (Second Edition)*, Addison-Wesley, 1989.
- [5] C. Cruz-Neira, D. J. Sandin, T. A. DeFanti, R. Kenyon, and J. C. Hart: “The CAVE, Audio Visual Experience Automatic Virtual Environment,” *Communications of the ACM*, pp. 64-72, 1992.
- [6] Carolina Cruz-Neira, Daniel J. Sandin, and Thomas A. DeFanti: “Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE,” *Proceedings of SIGGRAPH '93*, pp. 135-142, 1993.
- [7] Daniel B. Diner and Derek H. Fender: *Human Engineering in Stereoscopic Viewing Devices*, Plenum Press, 1993.
- [8] 江村 暁, 館 一 暲: “センサフュージョンを用いた頭部運動計測法,” 日本ロボット学会誌, Vol. 11, No. 6, pp. 929-932, 1993.
- [9] 江村 暁, 館 一 暲: “センサフュージョンによる実時間頭部運動計測法,” 日本電気学会論文誌 C 編, Vol. 115-C, No. 2, pp. 302-310, 1995.
- [10] 原島 博 (監修), 元木紀雄, 矢野澄男 (共編): “3 次元画像と人間の科学,” オーム社, 2000.
- [11] 橋本 渉, 岩田洋夫: “円面鏡を用いた球面没入型ディスプレイ: Ensphered Vision,” 日本バーチャルリアリティ学会論文誌, Vol. 4, No. 3, pp. 479-486, 1999.
- [12] 畑田豊彦: “疲れない立体ディスプレイを探る”, 日経エレクトロニクス, Vol. 444, pp. 205-223, 1988.
- [13] J. D. Hightower, E. H. Spain and R. W. Bowles: “Telepresence: A Hybrid Approach to High Performance Robots,” *Proceedings of the 3rd International Conference on Advanced Robotics (ICAR '87)*, pp. 563-573, 1987.
- [14] 平田亮吉, 星野 洋, 前田太郎, 館 一 暲: “人工現実感システムにおける物体形状を提示する力触覚ディスプレイ,” 日本バーチャルリアリティ学会論文集, Vol. 1, No. 1, pp. 23-32, 1996.

- [15] 廣瀬通孝, 木島竜吾, 佐藤洋一, 石井威望: “シースルー型 HMD を用いた仮想空間による実空間の修飾の研究,” 第 6 回ヒューマン・インタフェース・シンポジウム論文集, pp. 1–8, 1990.
- [16] Michitaka Hirose, Koichi Hirota, Ryugo Kijima, Masaaki Kanno, Kazuhiro Hayakawa, and Kensuke Yokoyama: “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, Vol. 1, No. 1, pp. 19–26, 1992.
- [17] 廣瀬通孝, 宮田亮介, 谷川智洋: “二次元実写画像を用いた三次元仮想空間の生成,” SICE Human Interface News & Report, Vol. 11, No. 2, pp. 209–214, 1996.
- [18] 廣瀬通孝, 小木哲朗, 石綿昌平, 山田俊郎: “没入型多面ディスプレイ (CABIN) の開発,” 日本バーチャルリアリティ学会第 2 回大会論文集, pp. 137–140, 1997.
- [19] 廣瀬通孝, 谷川智洋, 石田健亮: “嗅覚ディスプレイに関する研究,” 日本バーチャルリアリティ学会第 2 回大会論文集, pp. 155–158, 1997.
- [20] 廣瀬通孝, 小木哲朗, 石綿昌平, 山田俊郎: “多面型全天周ディスプレイ (CABIN) の開発とその特性評価,” 電子情報通信学会誌 D-II, Vol. J81-D-II, No. 5, pp. 888–896, 1998.
- [21] 廣瀬通孝, 小木哲朗, 山田俊郎, 玉川 憲, 金 成潤: “CABIN 間通信におけるビデオアバタの生成,” 第 9 回「人工現実感」研究会予稿集, pp. 43–46, 1999.
- [22] 星野 洋, 平田亮吉, 前田太郎, 舘 一 暉: “仮想触空間における物体形状の実時間提示法,” 日本ロボット学会誌, Vol. 15, No. 6, pp. 868–877, 1997.
- [23] L. F. Hodges and E. T. Davis: “Geometric Considerations for Stereoscopic Virtual Environments,” *Presence*, Vol.2, No.1, pp. 34–43, 1993.
- [24] Masahiko Inami, Naoki Kawakami, Dairoku Sekiguchi, Yasuyuki Yanagida, Taro Maeda, and Susumu Tachi: “Visuo-Haptic Display Using Head-Mounted Projector,” *Proceedings of IEEE Virtual Reality 2000*, New Brunswick, New Jersey, USA, pp. 233–240, 2000.
- [25] 泉 武博 (監修): “3 次元映像の基礎,” オーム社, 1995.
- [26] 川上直樹, 稲見昌彦, 柳田康幸, 前田太郎, 舘 一 暉: “オブジェクト指向型ディスプレイの研究,” 情報処理学会論文誌, Vol. 40, No. 6, pp. 2725–2733, 1999.
- [27] Mark J. Kilgard: *OpenGL™ Programming for the X Window System*, Addison-Wesley, 1996.
- [28] 高野英彦: “見やすさと距離認識誤差を考慮した仮想物体表示法の評価について,” 計測自動制御学会論文集, Vol. 34, No. 5, pp. 347–352, 1998.
- [29] W. Kruger, C.-A. Bohn, B. Frohlich, H. Schuth, W. Strauss, and G. Wesche: “The Responsive Workbench,” *IEEE Computer*, pp. 42–48, July 1995.
- [30] Leonard McMillan and Gary Bishop: “Plenoptic Modeling: An Image-Based Rendering System,” *Proceedings of SIGGRAPH 95*, pp. 39–46, Los Angeles, California, August 6–11, 1995.
- [31] 前田太郎, 荒井裕彦, 舘 一 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, Vol. 10, No. 5, pp. 655–665, 1992.

- [32] 前田太郎: “知覚系の視点から見た HMD,” 日本バーチャルリアリティ学会誌, Vol. 3, No. 2, pp. 11-17, 1998.
- [33] W. Mark, G. Bishop, L. McMillan: “Post-Rendering 3D Warping,” *Proceedings of 1997 Symposium on Interactive 3D Graphics*, pp. 7-16, 1997.
- [34] 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, 1989.
- [35] M. R. Mine: “Characterization of End-to-End Delays in Head-Mounted Display Systems,” *Tech. Rep. TR93-001*, Dept. of Computer Science, University of North Carolina at Chapel Hill, 1993.
- [36] 三津橋晃丈, 柳田康幸, 前田太郎, 舘 暲: “IPT 用カメラシステムの実装と評価,” 日本バーチャルリアリティ学会第4回大会論文集, pp. 161-164, 1999.
- [37] 苗村 健, 金子正秀, 原島 博: “実空間光線情報に基づくリアルな仮想空間の生成,” 日本バーチャルリアリティ学会第1回大会論文集, pp. 35-38, 1996.
- [38] J. Neider, T. Davis, and M. Woo: *OpenGL Programming Guide*, Addison-Wesley, 1993.
- [39] Yuichi Ohta and Hideyuki Tamura ed.: *Mixed Reality—Merging Real and Virtual Worlds*, Ohmsha, 1999.
- [40] 大石峰士, 舘 暲: “シースルー型 HMD における視覚パラメータの較正法,” 日本ロボット学会誌, Vol. 12, No. 6, pp. 911-918, 1994.
- [41] 大越孝敬: “三次元画像工学,” 朝倉書店, 1991.
- [42] 大倉典子, 大石峰士, 舘 暲: “シースルー型 HMD における視覚パラメータの距離に関する較正法,” 日本ロボット学会誌, Vol. 13, No. 4, pp. 580-582, 1995.
- [43] 大山英明, 常本直貴, 前田太郎, 舘 暲: “仮想環境へのテレイグジスタンスのための一手法,” 日本ロボット学会誌, Vol. 12, No. 8, pp. 1175-1182, 1994.
- [44] 大山 正, 今井省吾, 和気典二 編: 新編 感覚・知覚ハンドブック, 誠信書房, 1994.
- [45] 尾崎信之, 舘 暲: “透視変換下における仮想視点からの画像合成の一手法,” 計測自動制御学会論文集, Vol. 31, No. 12, pp. 2027-2029, 1995.
- [46] 通産省アールキューブ研究会編: “アールキューブ,” 日刊工業新聞社, 1996.
- [47] R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, and H. Fuchs: “The Office of the Future: A Unified Approach to Image-Based Modeling and Spatially Immersive Displays,” *Proceedings of SIGGRAPH '98*, pp. 179-188, 1998.
- [48] R. Raskar: “Oblique Projector Rendering on Planar Surfaces for a Tracked User,” *SIGGRAPH '99 Conference Abstracts and Applications*, p. 260, 1999.
- [49] W. Robinett and J. P. Rolland: “A Computational Model for the Stereoscopic Optics of a Head-Mounted Display,” *Presence*, Vol. 1, No. 1, pp. 45-62, 1992.

- [50] W. Robinett and R. Holloway: "The Visual Display Transformation for Virtual Reality," *Presence*, Vol. 4, No. 1, pp. 1-23, 1995.
- [51] J. P. Rolland and W. Gibson: "Towards Quantifying Depth and Size Perception in Virtual Environments," *Presence*, Vol. 4, No.1, pp. 24-49, 1995.
- [52] Sense8 Coporation: *World Toolkit Reference Manual*, 1993.
- [53] M. S. Shimamoto: "TeleOperator/TelePresence System (TOPS) Concept Verification Model (CVM) Development," *Recent Advances in Marine Science and Technology '92*, HI, USA, pp. 97-104, 1992.
- [54] 下条信輔: サブリミナル・マインド, 中央公論社, 1996.
- [55] *Graphics Library Programming Guide*, Silicon Graphics, Inc., 1991.
- [56] Ivan E. Sutherland: "The Ultimate Display", *Proceedings of the IFIPS Congress*, Vol. 2, pp. 506-508, 1965.
- [57] Ivan E. Sutherland: "A Head-Mounted Three-Dimensional Display," *Proc. Fall Joint Computer Conference, AFIPS Conf. Proc.*, Vol. 33, pp. 757-764, 1968.
- [58] 館 暲, 阿部 稔: "ト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, pp. 167-168, 1982.
- [59] S. Tachi, K. Tanie, K. Komoriya, and M. Kaneko: "Tele-existence (I): Design and Evaluation of a Visual Display with Sensation of Presence," *Proceedings of the 5th Symposium on Theory and Practice of Robots and Manipulators (RoManSy '84)*, pp. 245-254, 1984.
- [60] 館 暲, 荒井裕彦: "ト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, Vol. 7, No. 4, pp. 314-326, 1989.
- [61] Susumu Tachi, Hirohiko Arai and Taro Maeda: "Development of an Anthropomorphic Tele-Existence Slave Robot," *Proceedings of the International Conference on Advanced Mechatronics (ICAM)*, pp. 385-390, 1989.
- [62] Susumu Tachi, Hirohiko Arai, and Taro Maeda: "Tele-Existence Master-Slave System for Remote Manipulation," *Proceedings of International Workshop on Intelligent Robots and Systems (IROS '90)*, pp. 343-348, 1990.
- [63] 館 暲, 廣瀬通孝 (監修): "バーチャル・テック・ラボ," 工業調査会, 1992.
- [64] 館 暲: "人工現実感," 日刊工業新聞社, 1992.
- [65] 館 暲, 前田太郎: "人工現実感を有するト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, Vol. J75-D-II, No. 2, pp. 179-189, 1992.
- [66] Susumu Tachi and Ken-ichi Yasuda: "Evaluation Experiments of a Tele-Existence Manipulation System," *Presence*, Vol. 3, No. 1, pp. 35-44, 1994.

- [67] 館 暲, 安田賢一: “レイグジスタンス・マニピュレーション・システムの設計と評価,” 日本電気学会論文誌 C 編, Vol. 115-C, No. 2, pp. 172–181, 1995.
- [68] 谷村 透: “UNIX-LAN 構築の実践技術,” ソフト・リサーチ・センター, 1992.
- [69] 内海 章, Paul Milgram, 竹村治雄, 岸野文郎: “仮想空間表示における奥行き知覚誤差の要因について”, Human Interface News & Report, Vol. 9, No. 3, pp. 295–302, 1994.
- [70] Z. Wartell, L. F. Hodges, and W. Ribarsky: “Balancing Fusion, Image Depth, and Distortion in Stereoscopic Head-Tracked Displays,” *Proceedings of SIGGRAPH '99*, pp. 351–358, 1999.
- [71] 山田俊郎, 棚橋英樹, 小木哲朗, 廣瀬通孝: “完全没入型 6 面ディスプレイ COSMOS の開発と空間ナビゲーションにおける効果,” 日本バーチャルリアリティ学会論文誌, Vol. 4, No. 3, pp. 531–538, 1999.
- [72] 柳田康幸, 館 暲: “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, pp. 1087–1088, 1992.
- [73] Yasuyuki Yanagida and Susumu Tachi: “Virtual Reality System with Coherent Kinesthetic and Visual Sensation of Presence,” *Proceedings of the 1993 JSME International Conference on Advanced Mechatronics (ICAM '93)*, pp. 98–103, 1993.
- [74] 柳田康幸, 館 暲: “仮想環境作業システムのためのモデル構成法,” 第 32 回計測自動制御学会学術講演会予稿集, pp. 885–886, 1993.
- [75] Yasuyuki Yanagida and Susumu Tachi: “Coherency of Kinesthetic and Visual Sensation in Virtual Reality,” *Proceedings of the 1994 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (IEEE-MFI '94)*, Las Vegas, Nevada, USA, pp. 455–462, 1994.
- [76] 柳田康幸, 館 暲: “クラスを用いた仮想環境の構成法,” 日本電気学会論文誌 C 編, Vol. 115-C, No. 2, pp. 236–244, 1995.
- [77] 柳田康幸: “VR システムの構築方法—視覚による自然な 3 次元空間を構成するために—,” 第 4 回産業用バーチャルリアリティ展・セミナー要録, pp. 83–91, 1996.
- [78] Yasuyuki Yanagida and Susumu Tachi: “A Method for Smooth Task Execution Under Different Master/Slave Visual Parameters,” *Proceedings of the ICAT'96 (6th International Conference on Artificial Reality and Tele-Existence)*, Makuhari, Japan, pp. 65–72, 1996.
- [79] 柳田康幸, 稲見昌彦, 館 暲: “適切なオブティカルフローを実現する視覚提示法の研究 (第 1 報) —HMD 装着時の頭部回転運動に伴う遅延の補償手法—,” 日本バーチャルリアリティ学会第 2 回大会論文集, pp. 147–150, 1997.
- [80] 柳田康幸, 前田太郎, 館 暲: “CAVE-CABIN 型視覚提示装置用カメラシステムの設計,” 日本バーチャルリアリティ学会第 3 回大会論文集, pp. 149–152, 1998.
- [81] Yasuyuki Yanagida, Masahiko Inami and Susumu Tachi: “Improvement of Temporal Quality of HMD for Rotational Motion,” *Proceedings of the 7th IEEE International Workshop on Robot and Human Communication (RO-MAN'98)*, Takamatsu, Japan, pp. 121–126, 1998.

- [82] 柳田康幸, 前田太郎, 館 暲: “固定スクリーン型視覚提示装置を用いたテレイグジスタンス視覚系の構築手法,” 日本バーチャルリアリティ学会論文誌, Vol. 4, No. 3, pp. 539–547, 1999.
- [83] Yasuyuki Yanagida, Akitake Mitsuhashi, Taro Maeda and Susumu Tachi: “Implementation of Fixed-screen-based Telexistence Visual System,” *Proceedings of the ICAT '99 (9th International Conference on Artificial Reality and Telexistence)*, Tokyo, Japan, pp. 123–130, 1999.
- [84] Yasuyuki Yanagida, Taro Maeda and Susumu Tachi: “A Method of Constructing a Telexistence Visual System Using Fixed Screens,” *Proceedings of IEEE Virtual Reality 2000*, New Brunswick, New Jersey, USA, pp. 117–125, 2000.
- [85] 柳田康幸, 館 暲: “HMD を利用したテレイグジスタンスシステムにおける視覚・運動パラメータの整合性の影響,” 日本ロボット学会誌 (投稿中).
- [86] 横小路泰義, 菅原嘉彦, 吉川恒夫: “画像と加速度計を用いた HMD 上での映像の正確な重ね合わせ,” 日本バーチャルリアリティ学会論文誌, Vol. 4, No. 4, pp. 589–598, 1999.

関連発表

[誌上発表]

1. 柳田康幸, 館 暲: “クラスを用いた仮想環境の構成法,” 日本電気学会論文誌 C 編, Vol. 115-C, No. 2, pp. 236-244, 1995.
2. 柳田康幸, 前田太郎, 館 暲: “固定スクリーン型視覚提示装置を用いたトレイグジスタンス視覚系の構築手法,” 日本バーチャルリアリティ学会論文誌, Vol. 4, No. 3, pp. 539-547, 1999.
3. 柳田康幸, 館 暲: “HMD を利用したトレイグジスタンスシステムにおける視覚・運動パラメータの整合性の影響,” 日本ロボット学会誌 (投稿中).

[国際会議]

1. Yasuyuki Yanagida and Susumu Tachi: “Virtual Reality System with Coherent Kinesthetic and Visual Sensation of Presence,” *Proceedings of the 1993 JSME International Conference on Advanced Mechatronics (ICAM '93)*, Tokyo, Japan, pp. 98-103, 1993.
2. Yasuyuki Yanagida and Susumu Tachi: “Coherency of Kinesthetic and Visual Sensation in Virtual Reality,” *Proceedings of the 1994 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (IEEE-MFI '94)*, Las Vegas, Nevada, USA, pp. 455-462, 1994.
3. Yasuyuki Yanagida and Susumu Tachi: “A Method for Smooth Task Execution Under Different Master/Slave Visual Parameters,” *Proceedings of the ICAT'96 (6th International Conference on Artificial Reality and Tele-Existence)*, Makuhari, Japan, pp. 65-72, 1996.
4. Yasuyuki Yanagida, Masahiko Inami and Susumu Tachi: “Improvement of Temporal Quality of HMD for Rotational Motion,” *Proceedings of the 7th IEEE International Workshop on Robot and Human Communication (RO-MAN'98)*, Takamatsu, Japan, pp. 121-126, 1998.
5. Yasuyuki Yanagida, Akitake Mitsuhashi, Taro Maeda and Susumu Tachi: “Implementation of Fixed-screen-based Telexistence Visual System,” *Proceedings of the ICAT '99 (9th International Conference on Artificial Reality and Teleexistence)*, Tokyo, Japan, pp. 123-130, 1999.
6. Yasuyuki Yanagida, Taro Maeda and Susumu Tachi: “A Method of Constructing a Telexistence Visual System Using Fixed Screens,” *Proceedings of IEEE Virtual Reality 2000*, New Brunswick, New Jersey, USA, pp. 117-125, 2000.

[国内口頭発表]

1. 柳田康幸, 館 一 暲: “センサフュージョン機能を有する人工現実感システムの研究 (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, pp. 1087–1088, 1992.
2. 柳田康幸, 館 一 暲: “運動感覚と視覚を融合する人工現実感システム,” SICE Human Interface News and Report, Vol. 8, No. 3, pp. 325–330, 1993.
3. 柳田康幸, 館 一 暲: “仮想環境作業システムのためのモデル構成法,” 第 32 回計測自動制御学会学術講演会予稿集, pp. 885–886, 1993.
4. 柳田康幸, 館 一 暲: “センサフュージョン機能を有する人工現実感システムの研究 (第 2 報) — 運動感覚と視覚の融合 —,” 第 11 回日本ロボット学会学術講演会予稿集, pp. 1285–1286, 1993.
5. 柳田康幸, 館 一 暲: “運動感覚と視覚の融合の仮想環境へ及ぼす効果,” SICE Human Interface News and Report, Vol. 9, No. 4, pp. 467–472, 1994.
6. 柳田康幸, 館 一 暲: “センサフュージョン機能を有する人工現実感システムの研究 (第 3 報) — 仮想人間へのトレイグジスタンスにおける運動パラメータの実験的評価 —,” 第 12 回日本ロボット学会学術講演会予稿集, pp. 145–146, 1994.
7. 柳田康幸, 館 一 暲: “視覚パラメータの異なるトレイグジスタンスシステムにおける作業性改善の手法,” 日本機械学会 ロボティクス・メカトロニクス講演会 '97 講演論文集, Vol. B, pp. 789–792, 1997.
8. 柳田康幸, 稲見昌彦, 館 一 暲: “適切なオブティカルフローを実現する視覚提示法の研究 (第 1 報) — HMD 装着時の頭部回転運動に伴う遅延の補償手法 —,” 日本バーチャルリアリティ学会 第 2 回大会論文集, pp. 147–150, 1997.
9. 柳田康幸, 館 一 暲: “VR およびトレイグジスタンスにおける視覚パラメータの整合性に関する考察,” SICE Human Interface News & Report, Vol. 13, No. 2, pp. 207–212, 1998.
10. 柳田康幸, 前田太郎, 館 一 暲: “CAVE-CABIN 型視覚提示装置用カメラシステムの設計,” 日本バーチャルリアリティ学会第 3 回大会論文集, pp. 149–152, 1998.
11. 三津橋晃丈, 柳田康幸, 前田太郎, 館 一 暲: “IPT 用カメラシステムの実装と評価,” 日本バーチャルリアリティ学会第 4 回大会論文集, pp. 161–164, 1999.