

テレロボティクスにおける  
マンマシン・インタフェースの研究  
( Study on Man-Machine Interface of Telerobotics )

平成8年1月

井 上 清 一

学位論文：博士(工学) 甲51

# 目次

第1章 結 論	1
1.1 研究動向と問題点	1
1.2 本研究の目的と意義	4
1.3 本研究の内容梗概	6
第2章 複数のCCDカメラを用いた3次元計測	9
2.1 緒 言	9
2.2 3次元計測の原理とその手順	9
2.2.1 透視変換マトリクス	10
2.2.2 2つの透視図からの3次元情報の復元	12
2.3 透視変換マトリクスの測定	15
2.3.1 8点を用いた透視変換マトリクスの測定	15
2.3.2 16点を用いた透視変換マトリクスの測定	17
2.3.3 透視変換マトリクスの測定誤差	20
2.3.4 結果と考察	23
2.4 ヴィジュアルフィードバックによる3次元計測の理論	23
2.4.1 エンドイフェクターに取りつけたカメラによる3D位置計測の理論	24
2.4.2 マニピュレーターのリスト部(J4)の座標の求め方	24
2.4.3 エンドイフェクターに取りつけたカメラの透視変換マトリクスの求め方	25
2.4.4 ハンド系の3次元情報をロボット座標系に変換	27
2.5 ヴィジュアルフィードバックによる3次元計測の実験	29
2.5.1 エンドイフェクターに取りつけたカメラによる3D位置計測の実験	29
2.5.2 固定カメラから得られた3次元情報を用いた エンドイフェクターの姿勢制御	30
2.5.3 3次元計測実験の方法	32
2.5.4 3次元計測の実験結果	32
2.5.5 考 察	33
2.6 結 言	35

<b>第3章 仮想平面を用いた6自由度ハンドロボットの2次元制御</b> .....	36
3.1 緒言.....	36
3.2 遠隔操作システムの構成.....	36
3.2.1 遠隔操作システムの概要.....	37
3.2.2 立体視システム.....	40
3.3 仮想平面を用いたロボット制御.....	41
3.3.1 仮想平面の創成.....	41
3.3.2 仮想平面上の任意の点の2次元座標から3次元座標への変換.....	45
3.3.3 マウスを用いた入力装置.....	47
3.3.4 仮想平面を使ったマウスによる6自由度制御.....	48
3.3.5 ハンドロボットの制御.....	50
3.3.6 仮想平面の回転.....	51
3.4 結言.....	55
<b>第4章 仮想平面を用いた衝突回避と3次元空間の位置認識</b> .....	57
4.1 緒言.....	57
4.2 経路修正法による障害物回避.....	57
4.2.1 対象物体を掴んだエンドイフェクターの障害物回避.....	57
4.2.2 物体と障害物の接触条件と経路の修正.....	59
4.2.3 考察.....	61
4.3 仮想平面上に描かれた曲線を用いた障害物回避.....	61
4.3.1 仮想平面上に描かれた曲線の追跡.....	62
4.3.2 曲線追跡シミュレーション.....	63
4.3.3 実ロボットを使っての障害物回避.....	64
4.3.4 考察.....	64
4.4 立体視環境における3次元空間の位置認識.....	65
4.4.1 立体視環境における3次元空間の位置認識の実験.....	65
4.4.2 カーソルの立体表示の改善.....	67
4.4.3 位置認識実験の結果(実験条件#1、#2、#3).....	70
4.4.4 位置認識実験#4の実験方法.....	71
4.4.5 実験結果.....	73
4.4.6 考察.....	73

4.5	結 言	77
<b>第5章 仮想センシング法を用いた複数ハンドロボットの衝突回避</b>		<b>78</b>
5.1	緒 言	78
5.2	仮想センシング法のコンセプト	79
5.3	仮想センシング法の手順	80
5.4	衝突回避行動	85
5.5	考 察	87
5.6	結 論	88
<b>第6章 結 論</b>		<b>89</b>
参考文献		92
謝 辞		

## 第1章 緒論

### 1.1 研究動向と問題点

科学技術の進歩とともに、人間の健康と生存にとって危険な環境での作業が増加している。例えば、宇宙、深海、核施設、生物学または化学的に有害な環境、鉱山、建設現場、火事、警察または軍隊関連の作戦任務などがある。また、スケールの人間と整合しない超巨大な、または極微小な環境もたくさん考えられる。例えば、数十メートルの巨大な構造物の構築や、サブミクロンの原子操作の環境である。このような極限環境において、建設補修、資料採集、工作、人命救助などの多様な作業を実施するための、遠隔作業ロボット（テレロボティクス）の研究が進められている<sup>(1)(2)</sup>。

これらの研究には次のような歴史的経過が概観される<sup>(3)(4)</sup>。人間がいる場所から離れた作業現場にある機械の腕を器用に操る遠隔操作の概念はテレオペレータ (Teleoperator) と呼ばれた。これは第2次世界大戦後の原子力の利用とともに本格的な研究開発が始まった。通称「マジックハンド」と呼ばれるマスタースレイブ・マニピュレータがアルゴンヌ国立研究所を中心に研究開発された。レイ・ゲルツ (Ray Goertz) 博士を中心とするグループによる研究が進められ1948年には“ANL Model-1”と呼ばれる機械式の実用型のマスタースレイブ・マニピュレータが開発された。それを電気式に発展させた“ANL E1”は1954年に完成した。

続いて、米国における1960年代のテレオペレーションの研究は、エクゾスケルトン型人力増幅器が主流であった。これは、1950年代後半のGE (General Electric) 社のモッシャー (R. S. Mosher) 博士の研究に端を発するもので、60年代には米軍とGE社の共同プロジェクト「ハーディーマン (Hardyman)」に進展した。人間がエクゾスケルトンと呼ばれる外骨

格型の機械の鎧を着て、それにより器用さを失わずに自分の力を25倍に増幅し悪環境下の作業を遂行しようとするものであった。しかし、この方法は、以下に述べる大きな欠点を内包していたため多大の研究開発を行ったにも関わらず実用には至らなかった。すなわち、人間が装置の中に入り込むため、装置に故障があると人間の生命の危険に繋がる。一方、人間が装置の重要な部分を占めてしまうため装置を制御するコンピュータ、アクチュエータやセンサを配置する場所が限られ、絵には描けても実用的な実際のエグゾスケルトン型人力増幅器は構成できなかったのである。

テレオペレーション技術に自動化の手法を組み合わせた方法がスーパーバイザリーコントロール (Supervisory Control: 管理制御) で、1970年代にこの概念は広く様々の分野で利用されるようになった。この概念は1960年代後半、MITのトーマス・シェリダン (T. B. Sheridan) 教授によって提唱された<sup>(5)</sup><sup>(6)</sup>。管理制御では、人間は従来のマスタースレイブ方式のように、すべての動作を逐次送ることをやめ、ロボット言語による命令として伝達する。そのため、遠隔のロボットにローカルのコンピュータを配し、そのコンピュータは地上から送られたマクロな命令を解釈し、それを実行可能なシーケンスとして、ロボットのアクチュエータに働きかける。ロボットからのフィードバックも整理して、その結果を地上に送る。このような自律性をロボットに与えることにより、作業能率が向上する。

現在は、さらに自律性を重要視した知的テレオペレーションに発展している。また「テレロボティクス (Telerobotics)」という言葉も用いられるようになった。

従来のテレオペレータでは、人から機械に向かう制御系が強調されているが、機械から人間への感覚フィードバックは、対称型や力帰還型で力の情報をオペレータに伝える程度であり、人がその場で作業をしているような臨場感を持ちながら作業することは出来なかった。機械からオペレータへの感覚フィードバックを臨場感の高いものとするための研究が1980年代の初めに日本と米国で独立に生まれ近年世界的に盛んになりつつある。これは

「テレイグジスタンス (Teleexistence) 」 「テレプレゼンス (Telepresence) 」 ないしは「遠隔臨場制御」と呼ばれる新しい概念である。

遠隔臨場制御の概念を発展して、ロボットの制御に用いられれば、人力増幅機の中に人が入り込むような危険なことをしなくとも、人が中に入っているのと同等の感覚を確保しながら制御し、作業を行うことが出来るわけで、60年代の夢が今まさに実現しようとしている。

遠隔作業ロボットは、作業動作の源がどこにあるかによって2種類に分類される。一つは、人間の操縦動作のみを動作源とする操縦ロボット、他の一つは、人間の操縦動作とロボットによるプログラム動作の2つの動作源を有するテレロボットである。前者の操縦ロボットにおいては、人間にいかにか臨場感をもって作業状況を見せるかが研究の焦点となっている。

後者のテレロボットにおいては、人間とロボットのインタフェース、広い意味での対話技術が研究の焦点となっている<sup>(2)</sup>。

インタフェースの観点から、遠隔ロボットの問題点を捉えると次のことがあげられる。自動化や自律化によって、高性能化、高機能化したロボットは益々ブラックボックス化され、人間と機械との関係に新しい重大な問題が生じてきている。自動的に作動する運動機構の仕組みや構造、その結果得られる処理や速度などが、人間の理解力や感覚的レベルを越え始めているのである。人間はただ操作部への働きかけと、機械装置が処理した作業の結果を受け取るだけで、はじめから装置の内部はわからないものとして、想像することさえ諦めてしまうようになってきている。システムの中に人間が介在するテレオペレーションの目的は、予測出来なかった事態にシステムが陥った時、人間が直ちにその不測の事態に対処することにあつたはずである。

人間は理解できないものに対して心理的に不安感や不快感を抱くものであり、一種の拒

否反応を示すのは自然なことである。この心理的な問題は、遠隔ロボットシステムを設計する立場にある者として軽視出来ない重要な問題となっている。

もう一つの問題は、機械装置の多機能化による操作部分の複雑化である。多くの機能を発揮させるために、操作手順が複雑となり、これを記憶しておく必要が生じ、操作ミスを誘発しやすく、また使いにくくなっていることである<sup>(7)</sup>。

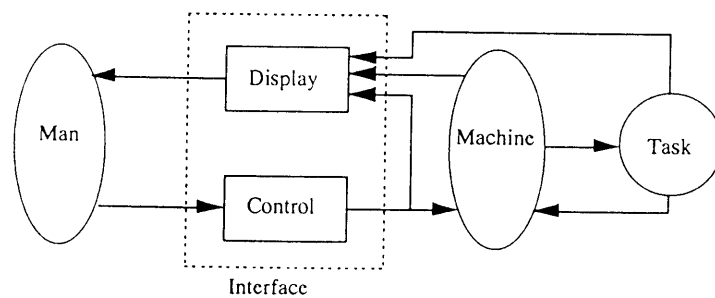
## 1.2 本研究の目的と意義

従来考えられてきた操作制御における人間とインタフェース部、および機械本体との関係は、以下のようなものになる。これらの関係を図示したものが図1.1(a)ある<sup>(7)</sup>。まず、人間とインタフェース部である操作部分との関係は次のようになっている。人間は、視覚や聴覚などの感覚器官により、メーター類の視覚情報やブザーなどの聴覚信号を認知し、操作部に与えるべき情報を判断する。ついで、ボタンやレバーなどの操作要素を指、手などの運動器官の機能により動かして、必要な物理的操作量を操作部に与える。

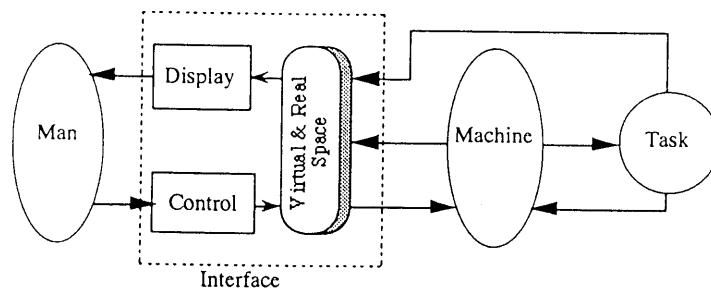
次に、インタフェース機能部分である操作部と、機械本体との関係について考えると、この部分は、人間から受け取った情報を、機械が作動反応をおこす情報に変換し、これを機械に伝達する部分と機械の作動状況を人間が理解できるものに変換して表示する部分に分けられる。従来考えられてきたインタフェース部は、操作要素と情報表示要素から構成されており、この2つの要素間の情報伝達はインタフェース部の外で操作要素から情報表示要素へに一方向に伝達されるものであった。この方式では、操作部の状況が視覚的に人間にフィードバックされない欠点を持っている。一例として、操作部のマスターマニピュレータを使ってハンドロボットを操作する場合を想定する。マスターマニピュレータとハンドロボットが同じ姿勢を保つ場合は問題が起きないが、作業のある部分で、ハンドロボ



ットが正反対の姿勢となった場合、人間にとって、ロボットを鏡を見ながら制御するような状況になり、オペレータに過大の負担を強いることになる。つまり、ハンドロボットがどのような姿勢にあっても、情報表示要素に呈示されているハンドロボットの空間的動作と、マスターマニピュレータの動作空間とが対応がとれていることが重要なものとなってくる。それを可能にするには操作部の状況も情報表示要素に呈示されるように、操作情報の処理もインタフェース部に組み込まなければならない<sup>(6)</sup>。



(a)



(b)

図1.1 人間とインタフェース部と機械本体との関係

図1.1 (b)は、本研究のコンセプトを説明したものである。図1.1 (a)との相違は、次の4点である。まず一つは、操作情報もインタフェース部に組み込み、操作情報を加工し、

人間の感覚にマッチした形態で人間に呈示していることである。2番目は、マシンの作動情報とマシンの出力情報を実空間情報として、また操作要素の出力である操作情報を仮想空間の中に描いた仮想情報として表し、実空間情報と仮想空間情報とをスーパーインポーズして人間に呈示していることである。3番目は、この呈示情報が立体視環境のもとで人間に呈示できることである。そして最後の特長は、インタフェース部が、マシンの作動情報を用いることによって、3次元情報の計測も可能にしていることである。

同様な研究として次の研究があげられる。実空間のロボットと仮想空間中の仮想の道具を重ね合わせて、ハンドロボットの制御に積極的に用いようとした試みとして、著者<sup>(9)</sup>の他に橋本や<sup>(10)</sup> L. B. Rosenbergの研究<sup>(11)</sup>がある。また、実世界と仮想世界を重ね合わせた空間の中で、仮想のポインターや仮想のスケールを導入して、人間とロボットのコミュニケーションの改善を図ろうとする研究として、P. MilgramやD. Drasicの研究があげられる<sup>(12)-(14)</sup>。

テレロボットにおいて、ロボットの果たす役割は2つあると考えられている<sup>(2)</sup>。一つは作業を実施する役割であり、後の一つは人間と機械とのインタフェースの役割である。この観点に立てば、テレロボットは作業をするロボットと人間とのインタフェースロボットの2台のロボットから構成されていて、それらが人間と向き合っているロボットシステムであると言い換えることが出来る。つまり、本研究においては、インタフェース部をもインタフェースロボットと捉えていることに特長がある。

### 1.3 本研究の内容梗概

本論文は、本章を含めて6章からなる。

第2章では、まずロボットが作業する環境内の3次元計測の原理と手順について述べて

いる。本研究で構築したシステムは、2台のカメラから得られた2つの映像から、人間に呈示される作業環境の任意の位置の3次元情報を計測することが出来る<sup>(9)</sup>。次に、実世界と仮想世界を対応させるために必要な透視変換マトリクスの各要素の測定とその誤差の検討を行っている。実空間の中の少なくとも既知6点が与えられれば、透視変換マトリクスの全ての要素を求めることが出来るが、実空間と仮想空間の良好な対応を実現するために、この章では、8点と16点を与えた場合の透視変換マトリクスの測定とその誤差について検討している。

つづいて、ヴィジュアルフィードバックによる3次元計測について述べている<sup>(16)</sup>。立体視環境のもとで3次元計測を行う場合、立体視を良好に与えるためには、2台のカメラを人間の両目の間隔とほぼ等しく保つ必要があり、一方、3次元計測の精度を向上させるためには、2台のカメラの間隔を出来るだけ広くとる必要がある。この相反する条件を克服するために、本章では、4台のカメラを用いたヴィジュアルフィードバックによる3次元計測を提案している。作業環境を広域に撮影する2台の固定カメラと計測精度を向上させるためエンドイフェクターに取り付けた2台のカメラを用いることによって、立体環境の呈示と3次元計測が同時にしかも良好に行えるようになったことを明らかにしている。

第3章では、仮想平面を用いた6自由度ハンドロボットの2次元制御について述べている。まず、ハンドロボットを遠隔操作するためのシステムの構成と立体視システムについて述べている。続いて、実空間と対応のとれた仮想空間の中に描かれた仮想平面と2次元入力装置であるマウスだけを用いて行う、6自由度のハンドロボットの遠隔制御の手法を提案している<sup>(15)</sup>。

第4章では、人間が与えた概略的回避経路を逐次修正しながら障害物回避経路を求める経路修正法による障害物回避と仮想平面上に描かれた曲線を追跡して行う障害物回避の方法を提案している<sup>(16)</sup>。さらに、立体視における3次元空間の位置認識について、実験的

に検討している。本研究では、時分割液晶眼鏡を使用して、オペレータに立体視環境を提供することが出来るようになったので、その立体視システムを用いて、ロボットの遠隔操作を行うとき、人間の位置認識にどの程度の改善がなされたかを実験的に検討している。

第5章では、仮想センシング法を用いた複数ハンドロボットの衝突回避について述べている。作業環境の中で、人間は両腕を巧みに用いて仕事をしているが、ロボットが複数で作業を行う場合、ロボット間の衝突回避が重要な問題となる。この章では、仮想センシング法と名付ける新しい方法を提案している。この方法の特長は、人間が本来行ってきた視覚的判定法を応用したもので、可能な限り数学的方法を避けようとするところにある<sup>(17)</sup>  
(18)。

第6章では、第2章から第5章で得られた成果について要約している。

## 第2章 複数のCCDカメラを用いた3次元計測

### 2.1 緒言

ロボットの遠隔制御を行う場合、ロボットの作業空間における対象物の3次元計測が不可欠となる。CCDカメラから送られてくる映像だけを手がかりにして、オペレーターは遠く隔たった場所にある環境を認識しながら、ロボットを操作することになる。本研究で製作した3次元情報計測システムは、対象物の位置をマウスで指定するだけで、その物体の3次元情報が求められる。また、この3次元計測システムは、計測に必要な全ての作業が、遠隔操作で行われることに特長がある。つまり、カメラの位置、姿勢が何らかの原因で変化しても、対象物及びハンドロボットを映像の視野の中に入れることだけが出来れば、3次元の位置計測が可能である。まずこのシステムを実現するための3次元計測の原理と手順について述べている<sup>(15)(19)(20)(26)(27)(31)</sup>。

つぎに、本研究で提案している計測法の根幹をなす透視変換マトリクスの測定について述べている。透視変換マトリクス測定の精度を向上させるために、既知点の増加による測定精度の向上について数値実験を行っている<sup>(25)</sup>。

最後は、4台のCCDカメラを用いたヴィジュアルフィードバックによる3次元計測について、その理論と実験結果について述べている<sup>(20)</sup>。

### 2.2 3次元計測の原理とその手順

本研究において製作したシステムは、3次元計測システムを搭載している。2台のCCDカメラを用い、得られた2つの映像から呈示される作業空間の中の任意の位置の3次元

情報を計測することが出来る。この計測システムにおいては、カメラから得られる2次元映像とロボットが作業する3次元の実作業空間の間の関係を表す透視変換マトリクスが既知であることが前提となる。

さらに、本システムにおいては、カメラからの映像とコンピュータグラフィックスがスーパーインポーズされ、奥行きのあるグラフィックスを用いて補うことも可能である。スーパーインポーズされるグラフィックスは、カメラから得られる映像の世界と空間的に一致している。例えば、映像の中に立方体が映っていると仮定する。この立方体の座標が与えられ、立方体をグラフィックスを用いて描くと現実世界の立方体とグラフィックスで描かれた仮想の立方体は、完全に一致させることが出来る。これらのグラフィックスの表示は、カメラに映し出される2次元映像とハンドロボットが作業する3次元空間との関係をあらず透視変換マトリクスを利用して呈示されるのである。

この章では、3次元計測に必要な基本的理論について述べている。以下の節では、まず透視変換マトリクスの求め方、つづいて、2つの透視図（TVカメラから得られる映像）を用いた3次元情報の復元の手法について説明している。

次に、マウスを用いた時の2次元入力による3次元制御のために必要な仮想空間に設定した仮想平面の創成法とそれを用いたときの3次元計測法について述べる。

### 2.2.1 透視変換マトリクス

3次元の立体空間と2次元の平面空間との対応関係を表す視変換マトリクスについて述べる。

図2.1にあるような3次元空間のある点の座標を $(X, Y, Z)$ とし、透視変換マトリクスを $T$ とすると、ディスプレイ上の点 $(X^*, Y^*)$ は次のようになる。

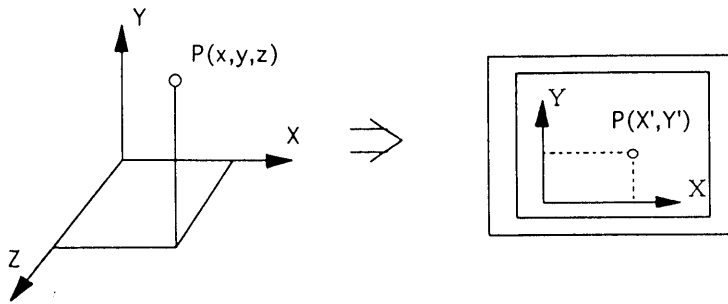


図 2. 1 3次元空間から2次元空間へ

$$\begin{bmatrix} X & Y & Z & 1 \end{bmatrix} \begin{bmatrix} T_{11} & T_{12} & 0 & T_{11} \\ T_{21} & T_{22} & 0 & T_{12} \\ T_{31} & T_{32} & 0 & T_{13} \\ T_{41} & T_{42} & 0 & T_{14} \end{bmatrix} \begin{bmatrix} X & Y & 0 & H \end{bmatrix} \quad (2.1)$$

$$X^* = X / H \quad , \quad Y^* = Y / H \quad (2.2)$$

この座標変換マトリクスが得られれば、3次元座標を2次元座標に変換可能となる。つまり、ディスプレイに映された映像と対応のとれたグラフィックを描くことが出来る。

更に、後で述べる3次元測定にも重要な役割を果たすものである。

次に、この透視変換マトリクスの求め方について述べる。

いま、式(2.1)、(2.2)を展開すると、

$$T_{11}X + T_{21}Y + T_{31}Z + T_{41} = HX^* \quad (2.3)$$

$$T_{12}X + T_{22}Y + T_{32}Z + T_{42} = HY^* \quad (2.4)$$

$$T_{14}X + T_{24}Y + T_{34}Z + T_{44} = H \quad (2.5)$$

のようになる。

ここで、式(2.5)を式(2.3), (2.4) に代入すると次のようになる。

$$(T_{11} - T_{14}X^*)X + (T_{21} - T_{24}X^*)Y + (T_{31} - T_{34}X^*)Z + (T_{41} - T_{44}X^*) = 0 \quad (2.6)$$

$$(T_{12} - T_{14}Y^*)X + (T_{22} - T_{24}Y^*)Y + (T_{32} - T_{34}Y^*)Z + (T_{41} - T_{44}Y^*) = 0 \quad (2.7)$$

さらに、式(2.6), (2.7)を、透視変換マトリクスの要素  $T_{ij}$  についてまとめると、

$$XT_{11} + YT_{21} + ZT_{31} + T_{41} - XX^*T_{14} - YX^*T_{24} - ZX^*T_{34} - X^*T_{44} = 0 \quad (2.8)$$

$$XT_{12} + YT_{22} + ZT_{32} + T_{42} - XX^*T_{14} - YX^*T_{24} - ZX^*T_{34} - X^*T_{44} = 0 \quad (2.9)$$

のようになり、3次元空間のある点の座標を  $(X, Y, Z)$  とし、その点のディスプレイ上の点を  $(X^*, Y^*)$  とし、これらを既知であるとする、式(2.8), (2.9)は12個の未知なる変換マトリクス要素  $T_{ij}$  についての2つの方程式を表していることになる。既知の点が1点あれば2つの方程式が導かれるので、既知の点を6点に増やすことにより12個の未知数を含む12個の方程式が得られ、この連立方程式を解くことにより、マトリクスの要素  $T_{ij}$  が求められ、変換マトリクスが既知となる。

### 2.2.2 2つの透視図からの3次元情報の復元

2台のCCDカメラから映し出される映像を使って、その映像の中にある任意の物体の



3次元座標を求めることが出来る。2台のCCDカメラからの映像は、スーパーインポーズにより重ねられモニターの画面に出力され、立体視スコープと連動することによって立体視が可能になる。これにより、3次元情報の復元ができる。なお、立体視システムの詳細は第3章で述べる。例えばある障害物の位置の座標を知りたいときは、その物体の両映像の画面の中で対応するポイントをマウスで指示することによって得られる。

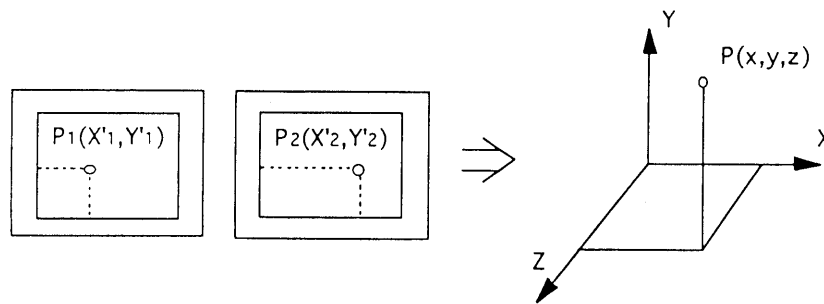


図2.2 2つの映像を使った3次元計測

図2.2は、2つの映像から3次元情報を復元する方法を説明したものである。ここでは、2つの映像を使った3次元計測の詳細について述べる。まず、この測定においては、透視変換マトリクスが分かっていることが前提である。その求め方の詳細は前節2.2 透視変換マトリクスで述べた通りである。

いま、式(2.6), (2.7)において、 $T_{ij}$ と  $X^*, Y^*$ が与えられた場合、未知数は空間座標  $X, Y, Z$ の3つであるので、この連立方程式を解くことはできない。しかし、もう一つ別のカメラより得られる  $T_{ij}$ と  $X^*, Y^*$ が与えられれば、 $X, Y, Z$ を求めることが出来る。つまり1つのカメラからの映像が得られれば、式(2.10), (2.11)が得られ、もう一方のカメラの映像が得られれば式(2.12), (2.13)が得られる。

$$(T_{11}^1 - T_{14}^1 X^{*1})X + (T_{21}^1 - T_{24}^1 X^{*1})Y + (T_{31}^1 - T_{34}^1 X^{*1})Z + (T_{41}^1 - T_{44}^1 X^{*1}) = 0 \quad (2.10)$$

$$(T_{11}^1 - T_{14}^1 X^{*1})X + (T_{22}^1 - T_{24}^1 X^{*1})Y + (T_{32}^1 - T_{34}^1 X^{*1})Z + (T_{42}^1 - T_{44}^1 X^{*1}) = 0 \quad (2.11)$$

$$(T_{11}^2 - T_{14}^1 X^{*2})X + (T_{21}^2 - T_{24}^2 X^{*2})Y + (T_{31}^2 - T_{34}^2 X^{*2})Z + (T_{41}^2 - T_{44}^2 X^{*2}) = 0 \quad (2.12)$$

$$(T_{11}^2 - T_{14}^1 X^{*2})X + (T_{22}^2 - T_{24}^2 X^{*2})Y + (T_{32}^2 - T_{34}^2 X^{*2})Z + (T_{42}^2 - T_{44}^2 X^{*2}) = 0 \quad (2.13)$$

上式をマトリクス形に書き換えると

$$[A] [X] = [B] \quad (2.14)$$

ここに

$$[A] = \begin{pmatrix} T_{11}^1 - T_{14}^1 X^{*1} & T_{21}^1 - T_{24}^1 X^{*1} & T_{31}^1 - T_{34}^1 X^{*1} \\ T_{11}^1 - T_{14}^1 Y^{*1} & T_{22}^1 - T_{24}^1 Y^{*1} & T_{32}^1 - T_{34}^1 Y^{*1} \\ T_{11}^2 - T_{14}^2 X^{*2} & T_{21}^2 - T_{24}^2 X^{*2} & T_{31}^2 - T_{34}^2 X^{*2} \\ T_{11}^2 - T_{14}^2 Y^{*2} & T_{22}^2 - T_{24}^2 Y^{*2} & T_{32}^2 - T_{34}^2 Y^{*2} \end{pmatrix} \quad (2.15)$$

$$[X]^T = [X \ Y \ Z] \quad (2.16)$$

$$[B]^T = [T_{44}^1 X^{*1} - T_{41}^1 \ T_{44}^1 Y^{*1} - T_{42}^1 \ T_{44}^2 X^{*2} - T_{41}^2 \ T_{44}^2 Y^{*2} - T_{42}^2] \quad (2.17)$$

ここで、式(2.14)は4つの方程式を表し、未知数  $x, y, z$  は3つであるので最小二乗法を用いて解くことにする。

$$[A]^T [A] [X] = [A]^T [B] \quad (2.18)$$

$$[X] = [A^T \cdot A]^{-1} [A]^T [B] \quad (2.19)$$

したがって、式(2.19)によって、変換マトリクス  $T_1, T_2$  と  $Z=0$  の平面に投影された物体の座標  $X^{*1}, Y^{*1}, X^{*2}, Y^{*2}$  が与えられれば、それに対応する物体の3次元座標  $X, Y, Z$  が求められることが分かる。

## 2. 3 透視変換マトリクスの測定

本研究では、CCDカメラに映った実世界とグラフィックで描かれた3次元の仮想世界を透視変換マトリクスを用いて対応をとっている。また、3次元計測においては、この透視変換マトリクスを用いて、2次元情報から3次元情報に復元をしている。したがって、透視変換マトリクスの測定は、立体視環境の呈示や3次元情報計測のもっとも基本的な技術となっている。

透視変換マトリクスを測定するには、実世界の既知の3次元座標  $P_i(x_i, y_i, z_i)$  ( $i=1\sim6$ ) と2次元映像の中で点  $P_i$ に対応する2次元座標  $p_i(x^*_i, y^*_i)$  ( $i=1\sim6$ )が最低必要となる。しかし、6点の既知点だけで透視変換マトリクスを求めた場合、3次元空間の座標系とCCDカメラの位置と姿勢によっては、精度よく透視変換マトリクスを計測できないことが、しばしば見受けられた。そこで、既知点を8点と16点与えた場合のマトリクス計測における誤差について検討を行った。

### 2.3.1 8点を用いた透視変換マトリクスの測定

図2.3は既知8点を用いた透視変換マトリクスの測定に関する説明をしたものである。左の図は、3次元空間の中の立方体を、右の図は、ディスプレイに表示された平面の映像の中の立方体を表している。式(2.20)の中の  $(X_i, Y_i, Z_i)$  は左の図中の“1”の3次元座標を表し、そして  $(x^*_i, y^*_i)$  は右の図の“1”の2次元座標を表している。この8点法においては、図2.3の外側の立方体の頂点8点を用いている。既知の値として立方体の頂点(8点)を与えることにより12個の未知数を含む16個の方程式が得られる。これを行列の形で表すと式(2.20)のようになる。

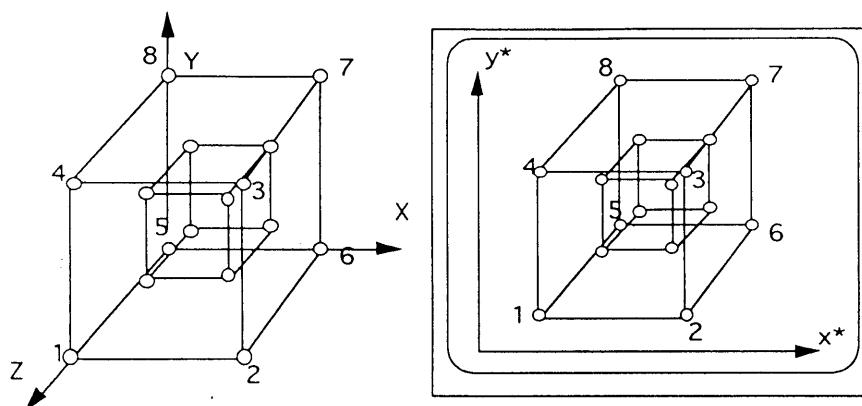


図 2.3 8点法と16点法の既知点

$$\begin{pmatrix}
 X_1 & 0 & -X_1 X_1^* & Y_1 & 0 & -Y_1 X_1^* & Z_1 & 0 & -Z_1 X_1^* & 1 & 0 & -X_1^* \\
 0 & X_1 & -X_1 Y_1^* & 0 & Y_1 & -Y_1 Y_1^* & 0 & Z_1 & -Z_1 Y_1^* & 0 & 1 & -Y_1^* \\
 X_2 & 0 & -X_2 X_2^* & Y_2 & 0 & -Y_2 X_2^* & Z_2 & 0 & -Z_2 X_2^* & 1 & 0 & -X_2^* \\
 0 & X_2 & -X_2 Y_2^* & 0 & Y_2 & -Y_2 Y_2^* & 0 & Z_2 & -Z_2 Y_2^* & 0 & 1 & -Y_2^* \\
 X_3 & 0 & -X_3 X_3^* & Y_3 & 0 & -Y_3 X_3^* & Z_3 & 0 & -Z_3 X_3^* & 1 & 0 & -X_3^* \\
 0 & X_3 & -X_3 Y_3^* & 0 & Y_3 & -Y_3 Y_3^* & 0 & Z_3 & -Z_3 Y_3^* & 0 & 1 & -Y_3^* \\
 X_4 & 0 & -X_4 X_4^* & Y_4 & 0 & -Y_4 X_4^* & Z_4 & 0 & -Z_4 X_4^* & 1 & 0 & -X_4^* \\
 0 & X_4 & -X_4 Y_4^* & 0 & Y_4 & -Y_4 Y_4^* & 0 & Z_4 & -Z_4 Y_4^* & 0 & 1 & -Y_4^* \\
 X_5 & 0 & -X_5 X_5^* & Y_5 & 0 & -Y_5 X_5^* & Z_5 & 0 & -Z_5 X_5^* & 1 & 0 & -X_5^* \\
 0 & X_5 & -X_5 Y_5^* & 0 & Y_5 & -Y_5 Y_5^* & 0 & Z_5 & -Z_5 Y_5^* & 0 & 1 & -Y_5^* \\
 X_6 & 0 & -X_6 X_6^* & Y_6 & 0 & -Y_6 X_6^* & Z_6 & 0 & -Z_6 X_6^* & 1 & 0 & -X_6^* \\
 0 & X_6 & -X_6 Y_6^* & 0 & Y_6 & -Y_6 Y_6^* & 0 & Z_6 & -Z_6 Y_6^* & 0 & 1 & -Y_6^* \\
 X_7 & 0 & -X_7 X_7^* & Y_7 & 0 & -Y_7 X_7^* & Z_7 & 0 & -Z_7 X_7^* & 1 & 0 & -X_7^* \\
 0 & X_7 & -X_7 Y_7^* & 0 & Y_7 & -Y_7 Y_7^* & 0 & Z_7 & -Z_7 Y_7^* & 0 & 1 & -Y_7^* \\
 X_8 & 0 & -X_8 X_8^* & Y_8 & 0 & -Y_8 X_8^* & Z_8 & 0 & -Z_8 X_8^* & 1 & 0 & -X_8^* \\
 0 & X_8 & -X_8 Y_8^* & 0 & Y_8 & -Y_8 Y_8^* & 0 & Z_8 & -Z_8 Y_8^* & 0 & 1 & -Y_8^*
 \end{pmatrix}
 \begin{pmatrix}
 T_{11} \\
 T_{12} \\
 T_{14} \\
 T_{21} \\
 T_{22} \\
 T_{24} \\
 T_{31} \\
 T_{32} \\
 T_{34} \\
 T_{41} \\
 T_{42} \\
 T_{44}
 \end{pmatrix}
 = 0 \quad (2.20)$$

ここで、式(2.20)を次のように変形する。

$$\begin{pmatrix}
 X_1 & 0 & -X_1 X_1^* & Y_1 & 0 & -Y_1 X_1^* & Z_1 & 0 & -Z_1 X_1^* & 1 & 0 \\
 0 & X_1 & -X_1 Y_1^* & 0 & Y_1 & -Y_1 Y_1^* & 0 & Z_1 & -Z_1 Y_1^* & 0 & 1 \\
 X_2 & 0 & -X_2 X_2^* & Y_2 & 0 & -Y_2 X_2^* & Z_2 & 0 & -Z_2 X_2^* & 1 & 0 \\
 0 & X_2 & -X_2 Y_2^* & 0 & Y_2 & -Y_2 Y_2^* & 0 & Z_2 & -Z_2 Y_2^* & 0 & 1 \\
 X_3 & 0 & -X_3 X_3^* & Y_3 & 0 & -Y_3 X_3^* & Z_3 & 0 & -Z_3 X_3^* & 1 & 0 \\
 0 & X_3 & -X_3 Y_3^* & 0 & Y_3 & -Y_3 Y_3^* & 0 & Z_3 & -Z_3 Y_3^* & 0 & 1 \\
 X_4 & 0 & -X_4 X_4^* & Y_4 & 0 & -Y_4 X_4^* & Z_4 & 0 & -Z_4 X_4^* & 1 & 0 \\
 0 & X_4 & -X_4 Y_4^* & 0 & Y_4 & -Y_4 Y_4^* & 0 & Z_4 & -Z_4 Y_4^* & 0 & 1 \\
 X_5 & 0 & -X_5 X_5^* & Y_5 & 0 & -Y_5 X_5^* & Z_5 & 0 & -Z_5 X_5^* & 1 & 0 \\
 0 & X_5 & -X_5 Y_5^* & 0 & Y_5 & -Y_5 Y_5^* & 0 & Z_5 & -Z_5 Y_5^* & 0 & 1 \\
 X_6 & 0 & -X_6 X_6^* & Y_6 & 0 & -Y_6 X_6^* & Z_6 & 0 & -Z_6 X_6^* & 1 & 0 \\
 0 & X_6 & -X_6 Y_6^* & 0 & Y_6 & -Y_6 Y_6^* & 0 & Z_6 & -Z_6 Y_6^* & 0 & 1 \\
 X_7 & 0 & -X_7 X_7^* & Y_7 & 0 & -Y_7 X_7^* & Z_7 & 0 & -Z_7 X_7^* & 1 & 0 \\
 0 & X_7 & -X_7 Y_7^* & 0 & Y_7 & -Y_7 Y_7^* & 0 & Z_7 & -Z_7 Y_7^* & 0 & 1 \\
 X_8 & 0 & -X_8 X_8^* & Y_8 & 0 & -Y_8 X_8^* & Z_8 & 0 & -Z_8 X_8^* & 1 & 0 \\
 0 & X_8 & -X_8 Y_8^* & 0 & Y_8 & -Y_8 Y_8^* & 0 & Z_8 & -Z_8 Y_8^* & 0 & 1
 \end{pmatrix}
 \begin{pmatrix}
 T_{11} \\
 T_{12} \\
 T_{14} \\
 T_{21} \\
 T_{22} \\
 T_{24} \\
 T_{31} \\
 T_{32} \\
 T_{34} \\
 T_{41} \\
 T_{42} \\
 T_{44}
 \end{pmatrix}
 =
 \begin{pmatrix}
 -X_1^* \\
 -Y_1^* \\
 -X_2^* \\
 -Y_2^* \\
 -X_3^* \\
 -Y_3^* \\
 -X_4^* \\
 -Y_4^* \\
 -X_5^* \\
 -Y_5^* \\
 -X_6^* \\
 -Y_6^* \\
 -X_7^* \\
 -Y_7^* \\
 -X_8^* \\
 -Y_8^*
 \end{pmatrix}
 \quad (2.21)$$

式(2.21)を最小二乗法を用いて  $A X = B$  という形にし、これを次に述べるLU分解で解く。それによって変換マトリクスが得られる。以後これを8点法と呼ぶことにする。

### 2.3.2 16点を用いた透視変換マトリクスの測定

次に、既知の点を16点与えて変換マトリクスを求めることを考える。与えられる既知の16点の位置は図2.3に示されている。8点法で説明した立方体とその内側にある立方体の頂点の合計16点を用いている。既知8点に加えて、12個の未知数を含む32の方程式が得られるので8点法同様、最小二乗法、LU分解により解くことにする。以後これ

を16点法と呼ぶことにする。

次にLU分解について説明する<sup>(23)</sup>。

$$A x = b$$

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ & \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \quad (2.22)$$

式(2.22)における係数行列AをLU分解法により下三角行列Lと上三角行列Uに分解する。

Aが正則行列であるから、

$$A = L U \quad (2.23)$$

と分解できる。ここで、LおよびUは、

$$L = \begin{pmatrix} f_{11} & & & & 0 \\ f_{21} & f_{22} & & & \\ & \cdots & \cdots & \cdots & \\ f_{n1} & f_{n2} & & & f_{nn} \end{pmatrix} \quad U = \begin{pmatrix} 1 & u_{12} & u_{13} & \cdots & u_{1n} \\ & 1 & u_{23} & \cdots & u_{2n} \\ & & \cdots & \cdots & \\ 0 & & & & 1 \end{pmatrix}$$

である。Uの対角要素はすべて1である。

式(2.23)を式(2.22)に代入して、

$$A x = L U x = b \quad (2.24)$$

が得られる。ここで補助ベクトルyを用いて、

$$U x = y$$

とおけば、式(2.24)は

$$L y = b \quad (2.25)$$

となる。式(2.25)より、代入法により順次 $y_i (i=1, 2, \dots, n)$ を求めることができる。

すなわち、

$$y_1 = b_1 / f_{11}$$

$$y_i = (b_i - \sum_{j=1}^{i-1} f_{ij} y_j) / f_{ii} \quad (i = 2, 3, \dots, n) \quad (2.26)$$

次に、求められた  $y_i$  を用いて、 $Ux = y$  より逆代入法で、 $x_i (i = n, n-1, \dots, 1)$  を解くことができる。

$$x_n = y_n$$

$$x_i = y_i - \sum_{j=i+1}^n u_{ij} x_j \quad (i = n-1, n-2, \dots, 1) \quad (2.27)$$

このように、係数行列  $A$  が、式(2.23) のように3角行列  $L$  と  $U$  に分解できれば、原式(2.22)を解くことができる。

ここで、どのようにして  $A$  を  $LU$  分解するのかについて述べる。

$A = LU$  より

$$A = \begin{pmatrix} f_{11} & f_{11}u_{12} & f_{11}u_{13} & \cdots & f_{11}u_{1n} \\ f_{21} & f_{21}u_{12}+f_{22} & f_{21}u_{13}+f_{22}u_{23} & \cdots & f_{21}u_{1n}+f_{22}u_{2n} \\ & & \dots\dots\dots & & \\ f_{n1} & f_{n1}u_{12}+f_{n2} & f_{n1}u_{13}+f_{n2}u_{23}+f_{n3} & \cdots & f_{n1}u_{1n}+f_{n2}u_{n2}+\cdots+f_{nn} \end{pmatrix}$$

が得られる。これより、

$$f_{11} = a_{11}, \quad f_{21} = a_{21}, \quad \dots, \quad f_{n1} = a_{n1}$$

および、

$$f_{11} u_{12} = a_{12} \quad \longrightarrow \quad u_{12} = a_{12} / f_{11}$$

$$f_{21} u_{12} + f_{22} = a_{22} \quad \longrightarrow \quad f_{22} = a_{22} - f_{21} u_{12}$$

.....

$$f_{n1} u_{12} + f_{n2} = a_{n2} \quad \longrightarrow \quad f_{n2} = a_{n2} - f_{n1} u_{12}$$

と順次、LとUの要素に変換できる。一般式は、

$$\left. \begin{aligned} u_{ij} &= (a_{ij} - \sum_{k=1}^{i-1} f_{ik} u_{kj}) / f_{ii} & (i < j) \\ f_{ij} &= a_{ij} - \sum_{k=1}^{j-1} f_{ik} u_{kj} & (i \geq j) \end{aligned} \right\} \quad (2.28)$$

である。

### 2.3.3 透視変換マトリクスの測定誤差

上述の8点法と16点法を用いてCCDカメラの姿勢に対応するx軸回り角度 $\theta$ とy軸回り角度 $\phi$ を15度おきに各々変化させて、マトリクス計測の誤差の検討を行った。

数値実験の手順は次の通りである。

(1)8点法については既知の透視変換マトリクスを用いて描いた立方体の頂点8点を取り、透視変換マトリクスを求める。また、16点法については8点法で用いた頂点8点に追加して、8点法で用いた立方体の内部に別の立方体を想定し、これらの8個の頂点も用いることとする。

$$\begin{aligned} 1 & ( 0, 0, 0 ) \\ 2 & ( 30, 0, 0 ) \\ 3 & ( 30, 0, 38 ) \\ 4 & ( 0, 0, 38 ) \\ 5 & ( 0, 40, 0 ) \\ 6 & ( 30, 40, 0 ) \\ 7 & ( 30, 40, 38 ) \\ 8 & ( 0, 40, 38 ) \end{aligned}$$

(2)測定された透視変換マトリクスの評価は、既知の透視変換マトリクスと求められた透視変換マトリクスの各要素の数値を比べるだけではできないので、次のような評価基準を



設定して行った。

30x40x38の立方体の頂点を既知のマトリクスと測定されたマトリクスを用いて、2次元座標に変換する。変換後の各頂点の座標が、どれだけずれているかを次式で計算し平均値をとることとした。

$$\text{err} = \frac{1}{n} \sum \{(x^{m_i} - x_i)^2 + (y^{m_i} - y_i)^2\}^{1/2} \quad (2.29)$$

$x_i, y_i$  ; 既知の透視変換マトリクスにより変換した座標

$x^{m_i}, y^{m_i}$  ; 求めた透視変換マトリクスにより変換した座標

$n$  ; 既知点の数

(3) これを $\theta \cdot \phi$ を各々0度から90度まで15度刻みで変化させて数値実験を行った。つまり $7 * 7 = 49$ の組み合わせのもとで調べた。その結果を表2.1に示す。ここで、誤差の値”1”は、ディスプレイの1ドットに相当する。つまり、真のマトリクスで描かれた頂点の位置と測定されたマトリクスで描かれた頂点の位置には1ドット分の誤差が生じることを表している。

表 2.1 測定された透視変換マトリクスの精度

$\theta$	$\phi$	err8	err16
0	0	0.310	0.251
0	15	0.606	0.328
0	30	0.646	0.258
0	45	0.732	0.312
0	60	0.561	0.328
0	75	0.551	0.326
0	90	0.376	0.314
15	0	0.526	0.236
15	15	0.932	0.361
15	30	0.635	0.353
15	45	0.465	0.282
15	60	0.823	0.230
15	75	0.783	0.418
15	90	0.941	0.336
30	0	0.795	0.217
30	15	0.812	0.373
30	30	0.653	0.339
30	45	0.658	0.344
30	60	0.924	0.459
30	75	0.851	0.476
30	90	0.747	0.312
45	0	0.737	0.402
45	15	0.886	0.343
45	30	0.957	0.442
45	45	0.659	0.356
45	60	0.753	0.318
45	75	0.687	0.304
45	90	0.708	0.328
60	0	0.394	0.255
60	15	0.751	0.369
60	30	0.813	0.452
60	45	0.505	0.248
60	60	0.811	0.402
60	75	0.795	0.433
60	90	0.733	0.402
75	0	0.608	0.371
75	15	0.715	0.351
75	30	0.991	0.407
75	45	0.935	0.301
75	60	1.054	0.462
75	75	0.785	0.365
75	90	0.897	0.452
90	0	0.360	0.300
90	15	0.793	0.395
90	30	0.598	0.376
90	45	0.595	0.378
90	60	0.883	0.420
90	75	0.708	0.295
90	90	0.376	0.314

#### 2.3.4 結果と考察

既知点16点を用いて透視変換マトリクスの各要素を求めた場合、座標軸の設定には、その影響はほとんど無いことが確かめられた。つまり、この場合は、どの方向から撮影しても精度の高い透視変換マトリクスが求められることが分かった。既知点8点を用いた場合、最小誤差と最大誤差の比較において、約3倍の差が生じていることが分かった。カメラの撮影方向に対して $\theta=0$ の時( $\phi=0$ または $\phi=90$ )、または $\theta=90$ の時( $\phi=0$ または $\phi=90$ )には、16点法と同レベルの低い誤差となっている。しかし、この状態にカメラを精度良く設定するには非常に時間がかかるので、16点法を用いた方が、既知点の入力に少々時間がかかるものの、カメラの姿勢に依存しない精度の良いマトリクスが求められるので、高精度の測定が必要なときは16点法を採用する方が良いことが分かった。

#### 2.4 ヴィジュアルフィードバックによる3次元計測の理論

この節では、エンドイフェクターに取りつけたカメラと固定カメラの両者を用いたヴィジュアルフィードバックによる3次元計測について述べている。立体視環境のもとで3次元計測を行う場合、立体視を与えるためには、2台のカメラを人間の両目の間隔とほぼ等しく保つことが必要である。しかし3次元計測の精度を向上させるためには、2台のカメラの間隔を出来るだけ広くとることが望ましい。この相反する条件を克服するために、本研究では、4台のカメラを用いたヴィジュアルフィードバックによる3次元計測を提案している。作業環境を広域に撮影する2台の固定カメラと計測精度を向上させるためのエンドイフェクターに取り付けた2台のカメラを用いている。合計4台のカメラを用いることで立体視環境の呈示と3次元計測が同時にしかも良好に行えるようになった。

#### 2.4.1 エンドイフェクターに取りつけたカメラによる3D位置計測の理論

エンドイフェクターに取り付けられたカメラによる3次元計測においては、エンドイフェクターを支える関節の角度の変化によって取り付けられたカメラが移動することになるので、その補正が必要となる。まず、1) エンドイフェクターの先端の座標 $J_6(x_6, y_6, z_6)$ から、エンドイフェクターを支える関節 $J_4(x_4, y_4, z_4)$ の座標を求める必要がある。続いて、2) 2台のカメラから得られるハンド座標系の3次元座標 $(x_h, y_h, z_h)$ を関節 $J_4$ を原点とする座標系に変換する。そして最後に、3)  $J_4$ を原点とする座標系をロボット座標系に変換する必要がある。

#### 2.4.2 マニピュレーターのリスト部 ( $J_4$ ) の座標の求め方

まず、 $J_4$ の座標  $(x_4, y_4, z_4)$  をエンドイフェクターの先端部の現在位置 $J_6(x_6, y_6, z_6, p, r)$ より求める。ただし、ここで $p, r$ はそれぞれピッチ角、ロール角を表す。また、 $J_6$ はハンドロボットのエンドイフェクターの先端座標を求めるコマンドより求められる。

図2.4において、ハンドの $x$ 軸との傾き角 $\phi$ は

$$\begin{aligned} |x| < 0.001 \quad \text{の時} & \quad \phi = 90 \\ x \geq 0 \quad \text{の時} & \quad \phi = \tan^{-1}(y_6/x_6) \\ x < 0 \quad \text{の時} & \quad \phi = 180 + \tan^{-1}(y_6/x_6) \end{aligned} \quad (2.30)$$

$J_4$ の座標  $(x_4, y_4, z_4)$  は

$$x_4 = x_6 - l \cos p \cos \phi \quad (2.31)$$

$$y_4 = y_6 - l \cos p \sin \phi \quad (2.32)$$

$$z_4 = z_6 - l \sin p \quad (2.33)$$

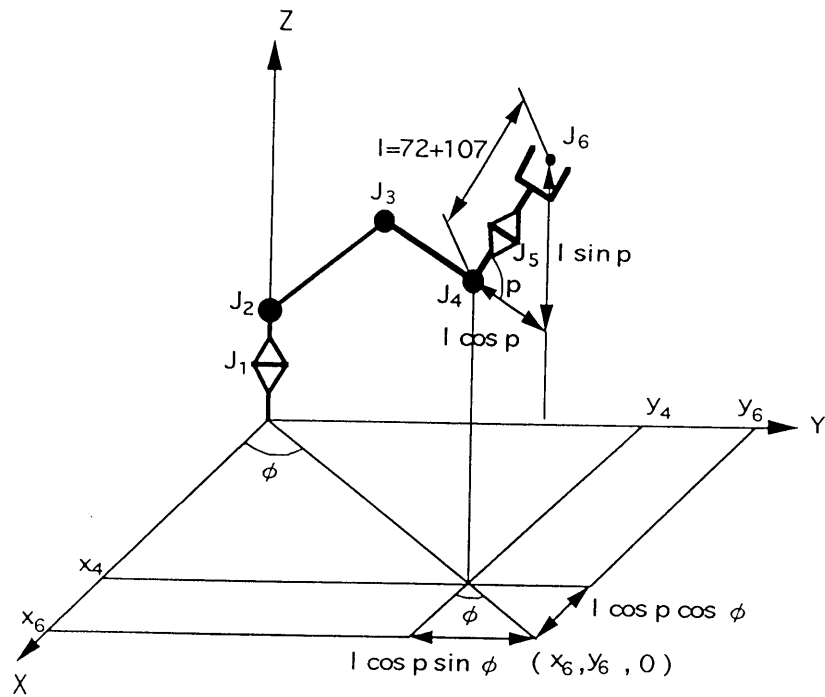


図 2.4 ロボット座標系

のように表される。ただし、 $l$  はエンドイフェクターの長さを表している。

本研究では  $l=179\text{mm}$  を採用している。

### 2.4.3 エンドイフェクターに取り付けたカメラの透視変換マトリクス of 求め方

カメラは図 2.5 の様に取り付けられている。点 1 ~ 6 の 3 次元座標が与えられれば、透視変換マトリクスは求められる。

ここでは、1 ~ 6 の既知 6 点の座標系をハンド座標系と呼ぶことにする。この系では 5, 6 の中点を座標系の原点とし、5, 6 を通る軸を  $x_h$  軸、エンドイフェクターの前方を  $y_h$  軸、

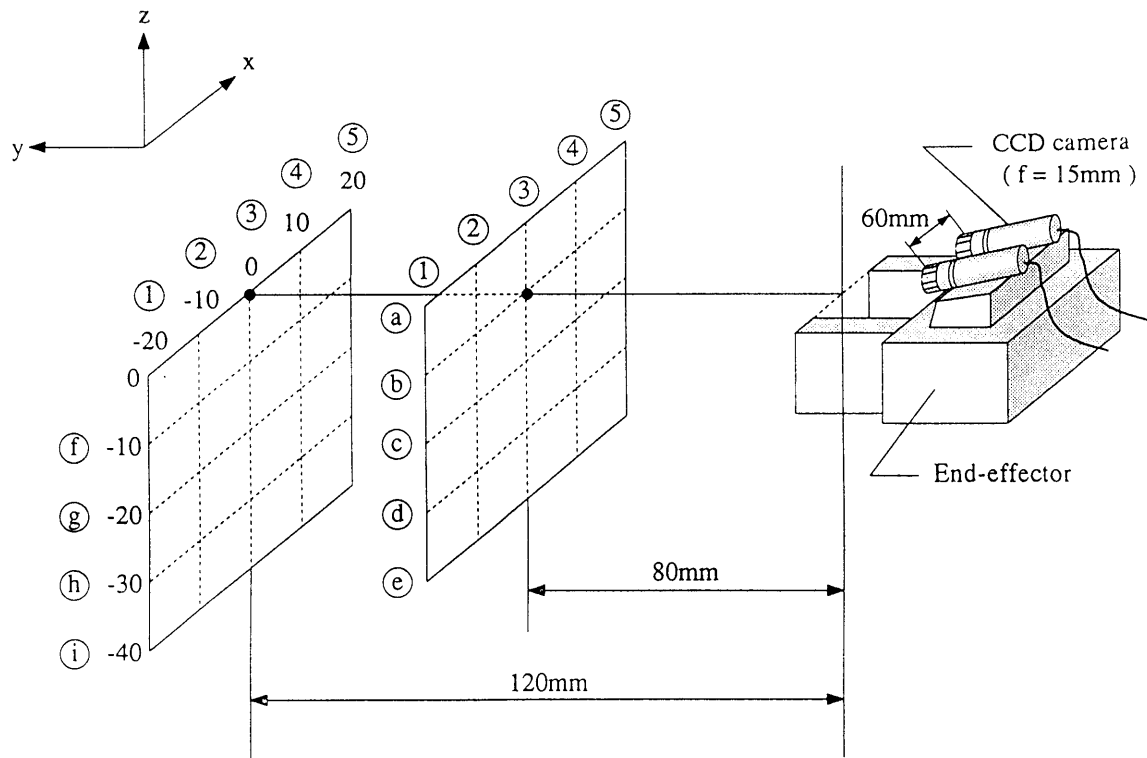


図 2.5 エンドイフェクターに取り付けたカメラ

その上方を $z_h$ 軸と呼ぶことにする。また、ハンド座標系の数値は以下の通りである。

	x	y	z	
1	-20	120	10	
2	20	120	10	
3	-20	120	-20	
4	20	120	-20	
5	-10	80	0	
6	10	80	0	単位 mm

2つの透視変換マトリクスが求められれば、エンドイフェクター上に取り付けられた2台のカメラを用いれば、対象物体のハンド座標系における3次元座標( $x_h, y_h, z_h$ )を求めることができる。求められた $x_h, y_h, z_h$ はエンドイフェクターの先端部を原点とする座

標系であるので、関節 $J_4$ を原点とするハンド座標系に変換する必要がある。

#### 2.4.4 ハンド系の3次元情報をロボット座標系に変換

##### 1) ハンド座標系を関節 $J_4$ を原点とする座標系に変換

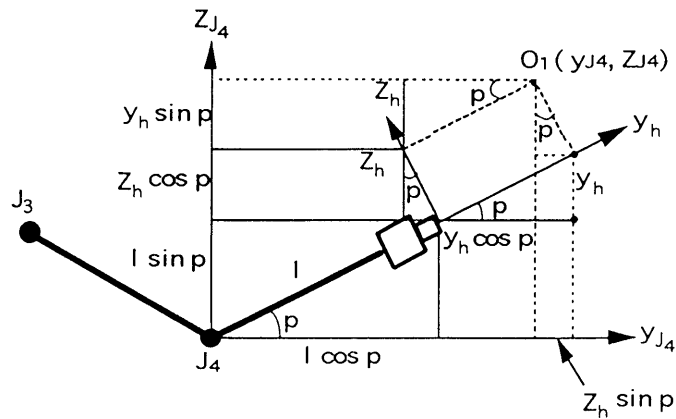


図 2.6  $J_4$ を原点とする座標系

$x_h$ 軸は $J_4$ の回転軸と平行であるから、変換の必要はないので

$$x_{J4} = x_h \quad (2.34)$$

となる。

$y_{J4}$ ,  $z_{J4}$ は図 2.6 より次のように求められる。

$$y_{J4} = l \cos p + y_h \cos p - z_h \sin p \quad (2.35)$$

$$z_{J4} = l \sin p + y_h \sin p + z_h \cos p \quad (2.36)$$

これらをまとめると

$$\begin{pmatrix} x_{J4} \\ y_{J4} \\ z_{J4} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos p & -\sin p \\ 0 & \sin p & \cos p \end{pmatrix} \begin{pmatrix} x_h \\ y_h \\ z_h \end{pmatrix} + \begin{pmatrix} 0 \\ l \cos p \\ l \sin p \end{pmatrix} \quad (2.37)$$

となる。よって、式(2.37)を用いると原点がJ4に移動した時の対象物の位置が求められる。

2) J4を原点とするハンド座標系をロボット座標系に変換

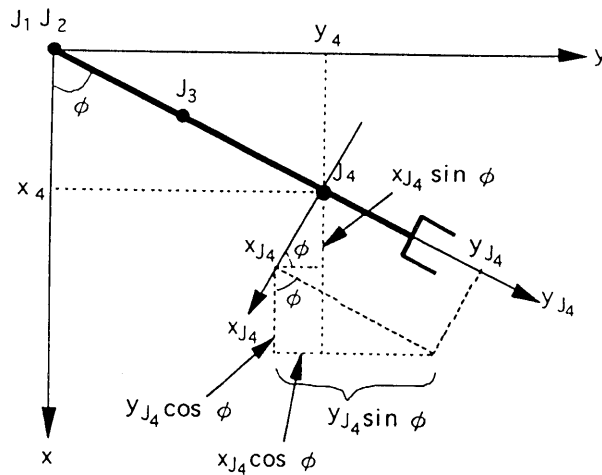


図 2.7 ハンド座標系をロボット座標系に変換

z軸方向は、両座標系のz軸が平行であるので回転による変換は不要となり、次の様になる

$$z = z_4 + z_{J4} \quad (2.38)$$



x, y は図 2.7 より

$$x = x_4 + x_{J4} \sin \phi + y_{J4} \cos \phi \quad (2.39)$$

$$y = y_4 - x_{J4} \cos \phi + y_{J4} \sin \phi \quad (2.40)$$

これらをまとめると

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \sin \phi & \cos \phi & 0 \\ -\cos \phi & \sin \phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{J4} \\ y_{J4} \\ z_{J4} \end{pmatrix} + \begin{pmatrix} x_4 \\ y_4 \\ z_4 \end{pmatrix} \quad (2.41)$$

となる。

式(2.31), (2.31), (2.33), (2.37), (2.41)を用いることにより、エンドイフェクターに取りつけたカメラから得られる3D情報をロボット座標系の3D情報に変換できる。したがって、x, y, z を与えることによってロボットのエンドイフェクターは物体を掴みに行くことが可能となる。

## 2.5 ヴィジュアルフィードバックによる3次元計測の実験

### 2.5.1 エンドイフェクターに取りつけたカメラによる3D位置計測の実験

#### 1) 実験方法

図 2.5 に示すようにカメラを搭載したエンドイフェクターの前方に計測用に作成した平板を80mmと120mmのところに配置し、それぞれについて任意の測定点で位置計測を行

う。なお、任意の点は図 2.5 の中で丸で囲まれた数字とアルファベットを座標とする所であり、平板内の数字はハンド座標系を示す。

## 2) 実験結果

実験の測定値と誤差（実際の測定点の位置の座標値と実験の測定値の差）を表 2.2 に示す。

表 2.2 3次元計測の精度

	1			3			5		
	x	y	z	x	y	z	x	y	z
a	-0.6	0.2	0	-0.6	0.7	-0.2	-0.6	0.7	-0.4
c	-0.6	0.5	-0.2	-0.6	0.8	-0.5	-0.4	0.9	-0.6
e	-0.8	0.4	-0.3	-0.5	0.3	-0.7	-0.5	0.5	-0.4
f	0.7	0.4	0.7	0.8	1.7	1.1	1.0	1.2	0.8
h	0.5	1.1	1.2	0.8	1.2	1.1	0.7	1.6	1.5
i	0.5	0.8	1.2	0.8	1.9	1.5	0.9	1.4	1.2

結果は距離 80 mm の時で誤差は 0.9 mm 以内になり、120 mm の時で 1.7 mm 以内になった。

また、x 方向の誤差が全体的に小さく y 方向の誤差が大きめである。

### 2.5.2 固定カメラから得られた 3次元情報を用いたエンドイフェクターの姿勢制御

この計測法においては、固定カメラで得られた 3次元情報をもとに、エンドイフェクターが、対称物体に接近し、続いてエンドイフェクターに取り付けられたカメラで 3次元計測を行う。すなわち、固定カメラで得られた 3次元情報をもとに対象物体に接近するわけ

であるが、接近したときにエンドイフェクターに取り付けられたカメラの視野に対象物体の映像が映るようにエンドイフェクターの姿勢を制御しなければならない。その制御法を説明したものが図2.8 (a), (b)である。(a)に示すように、対象物体とエンドイフェクターの先端部との距離を $L_0$ とする時、距離の指定だけではエンドイフェクターに取り付けたカメラの視野に対象物体が映る保証はない。(b)に示すようにエンドイフェクターの延長線上に、対象物体が来るようにエンドイフェクターの姿勢を制御する必要がある。

本研究で用いられたハンドロボットは、エンドイフェクターの先端の座標を指定することによって、エンドイフェクターがその位置へ移動することが出来る。通常の制御ではエンドイフェクターの長さを  $L$  mm と設定しているが、対象物体への接近に際しては、この長さを  $L + L_0$  と一時的に設定を変えている。このことによって、エンドイフェクターは、実際よりも長く設定され、あたかも  $L + L_0$  の長さを持っているものと解釈され対象物体

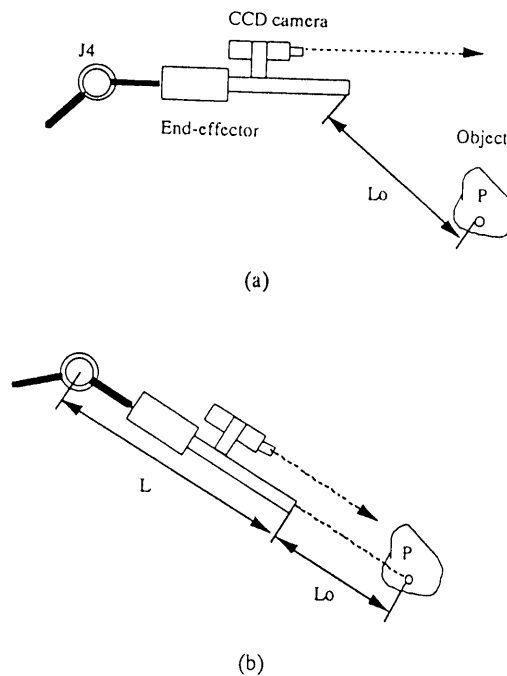


図2.8 エンドイフェクターの姿勢制御

を掴む姿勢が決定される。このことによって、エンドイフェクターの延長線上に対象物体が来るように制御され、エンドイフェクターに取り付けられたカメラの視野に対象物体が位置するようになり、2回目の3次元計測の準備が整うこととなる。

### 2.5.3 3次元計測実験の方法

この実験は高さ 135 mmの物体を任意の位置に置いて、固定カメラとエンドイフェクターに取り付けた移動カメラを用いて3次元計測を行い、その対象物を掴むものである。

目標となる物体の位置は、ハンドロボットの先端位置取得コマンドを用いて、あらかじめ測定をすませておく。

まず固定カメラから得られた3次元計測データを用いてロボットを目標物の近くへ移動させる。続いてエンドイフェクターに取り付けた移動カメラによる3次元計測データを用いて、物体を取りに行かせる。ロボットが物体を掴んだ位置は、ロボットの先端位置取得コマンドより得られる。あらかじめ測定しておいた物体の位置と物体をロボットが掴んだ位置の差を求めることにより、ヴィジュアルフィードバックによる3次元計測の精度を調べることが出来る。

### 2.5.4 3次元計測の実験結果

実験結果として物体の上端の位置座標  $(x, y, z)$  とロボットが物体を取りに行った位置の座標  $(x_r, y_r, z_r)$ 、並びに誤差(両者の差)  $(\Delta x, \Delta y, \Delta z)$  を表2.3に示す。誤差の平均と、その標準偏差は次のようになった。

表 2.3 ハンドロボットの移動精度 (単位 mm)

No.	物体の位置			ロボットの移動位置			誤差		
	x	y	z	$x_r$	$y_r$	$z_r$	$\Delta x$	$\Delta y$	$\Delta z$
1	-100	303	135	-95.44	322.55	128.85	-4.56	+19.55	-6.15
2	100	303	135	109.23	319.46	129.41	+9.23	+16.46	-5.59
3	0	453	135	10.21	471.79	132.88	+10.21	+18.79	-2.12
4	0	403	135	9.65	424.23	134.04	+9.65	+21.23	-0.96
5	-150	353	135	-147.74	377.40	133.52	-2.26	+24.40	-1.48
6	150	303	135	161.30	320.58	131.95	+11.30	+17.58	-3.05
7	0	503	135	11.45	520.15	129.86	+11.45	+17.15	-5.14
8	-100	253	135	-96.52	269.73	124.39	-3.48	+16.73	-10.61
9	150	303	135	157.96	315.06	127.57	+7.96	+12.06	-7.43
10	0	503	135	10.76	516.76	131.68	+10.76	+13.76	-3.32

x 方向で 6.02mm ( 6.29mm)

y 方向で 17.77mm ( 3.45mm)

z 方向で -4.59mm ( 2.85mm)

この値より、y 方向の誤差が約 18 mm あり最も大きい誤差となっている。しかし、標準偏差をみると 3 mm 程度ではばらつきが非常に小さいことが分かる。これは偏りによる誤差と考えられるので補正が可能である。z 方向は誤差も小さく、ばらつきも非常に小さいことが分かる。x 方向は、固定カメラの奥行き方向になるので、最も誤差のばらつきが大きくなっている。

### 2.5.5 考察

エンドイフェクターに取り付けたカメラによる実験結果では、測定誤差は最大で 1.9mm

であり、固定カメラとエンドイフェクター上のカメラの両方を用いたシステムの実験結果では測定誤差の最大は y 方向の 24.4 mm であった。

しかし、前節で述べたように y 方向の誤差は平均値で 17.77 mm と大きいですが、標準偏差は 3.45 mm とばらつきが非常に小さいことが分かる。これは、測定装置の設定の不備が原因と考えられる偏りによる誤差であると推定されるので、補正を行うことによって最大で 6 mm 程度まで減少させることが出来るであろう。

同様に、z 方向についても、標準偏差が 2.85 mm と非常に小さい値となっているために、測定装置の設定誤差が最も大きな誤差の要因と考えられる。従って、これも補正を行うことによって測定誤差を小さくすることが出来き、最大で 6 mm 程度の誤差にすることが出来ると考えられる。

x 方向の誤差は、プラスとマイナス両方向にばらついているので、測定装置の設定不備の他に、測定点へのポインティングによる誤差が大きく含まれていると考えられる。

今回の実験によって、全体的には、補正を行えば最大で 11.45 mm の測定誤差の範囲内で 3 次元測定が可能であることを示すことが出来た。

より良い精度を得るための今後の課題として、具体的な誤差の要因を考えてみることにする。まず、問題となる点は固定カメラは物体の近くにエンドイフェクターを寄せるためにしか理論的には使用していない。つまり、固定カメラはエンドイフェクターのカメラに物体を映せるように出来れば役割を果たし、誤差の要因とは成り得ないことである。このことを考慮すれば、1) 3次元計測データを得るために使うエンドイフェクター上に取り付けたカメラと実世界の関係を示す透視変換マトリクス精度、2) ハンド座標系からロボット座標系への座標変換精度、3) ロボットの駆動系の固有誤差、4) CCDカメラの設置環境が主な要因と考えることが出来る。

それぞれの要因の誤差に対する影響の度合いを考えると、ロボットの固有誤差は、取扱

説明書では使用したロボットの位置繰り返し精度は $\pm 0.3$  mm である。しかし、実際のロボットは使用回数が多いことや、エンドイフェクターに2台のCCDカメラを取り付けているので、精度が落ちている可能性があり、大きい影響はないとしても数mmの誤差の原因にはなっていると思われる。特に、誤差の値が同符号に偏っているy方向とz方向にその可能性がある。特に、エンドイフェクターに取り付けたカメラの姿勢が、ロボットの動作時に多少くいを生じている可能性が考えられる。

4台のカメラを用いた、2段階の計測法により、約10 mm 前後の誤差でハンドロボットが対象物を取りにいけることを実験的に確認することが出来た。エンドイフェクターに取り付けたカメラの固定法をもう少し堅固なものにすることが出来れば、さらに良い精度でハンドロボットを制御できるものと考えられる。

## 2. 6 結 言

CCDカメラを用いた3次元情報計測の原理と手順、透視変換マトリクスの測定、3次元計測実験について述べてきた。これらの結果より、作業環境から遠く隔たった所にいるオペレーターが、作業環境の中にある任意の対象物の3次元位置を良好に計測出来ることを実験的に示すことが出来た。このことより、ハンドロボットで遠隔操作をする場合、呈示された映像を見ながらハンドロボットのエンドイフェクターを逐次近づけるのではなく、得られた3次元情報をもとに直接対象物を掴む動作が可能なり、オペレーターの負担を軽減することが可能となった。

## 第3章 仮想平面を用いた6自由度ハンドロボットの2次元制御

### 3.1 緒言

本章では、ハンドロボットが作業する実空間と、グラフィックスにより呈示される仮想空間をスーパーポーズした作業空間の中で、仮想平面を用いた2次元制御と障害物回避を行う方法について提案している。このシステムにおいては、人間オペレータは、実空間と対応のとれた仮想空間の中に描かれた仮想平面と2次元入力装置であるマウスだけを用いて6自由度のハンドロボットを遠隔制御することができる<sup>(15)(21)</sup>。つまり、仮想平面上の任意の点をマウスで指定するだけで、ハンドロボットの先端効果器の3次元位置とピッチ角をまず制御することができる。先端効果器は仮想平面上だけを移動するように拘束されているが、この仮想平面を任意の方向に回転させることによって、先端効果器を作業空間の任意の位置に移動させることが可能である。先端効果器の姿勢とハンドの開閉は、仮想空間の中に設置されたボタンを操作することによって制御できる。これらのボタンの位置はハンドロボットが移動しても、たえずハンドロボットのリスト部の所に提示されるので、操作性の向上に非常に貢献している。

### 3.2 遠隔操作システムの構成

本研究で開発した遠隔操作システムは、入力装置としてコンピュータの2次元入力装置であるマウスを使用していることに特長がある。マウスとキーボードだけで全ての遠隔操作が可能であるので、このシステムは携帯性と経済性に富むものとなっている。さらに、人間には立体視環境が与えられるので、従来の2次元ディスプレイの欠点であった奥行き



方向の欠如による欠陥を補うことが出来る。また、この立体視システムは、家庭用のビデオデッキを用いて作業環境や作業状態を記録することもできるので、不測の事態に陥った場合の状況を立体視環境のもとで記録再現が可能となり、事故発生の原因究明にも役立つことが出来る。

### 3.2.1 遠隔操作システムの概要

今回開発したシステムの概要を図3.1に示す。テレオペレータとして6自由度（ハンドの開閉を含む）のハンドロボットを用い、このロボットを人間オペレータが立体視映像の提示のもとで、マウスを用いて遠隔制御することを想定している。

この遠隔操作システムは、次に述べる5つの特長を持っている。

- 1) 4台のCCDカメラを用いたヴィジュアルフィードバック3次元計測システム。
- 2) 作業環境内のロボットを立体視環境のもとで制御可能である。
- 3) 仮想平面や仮想操作ボタンを表示する仮想空間が、ロボットが作業する実世界と空間的に対応している。
- 4) 入力装置は、2次元の入力装置であるマウスを用い、操作性を高めている。
- 5) 3次元計測装置と仮想空間に描かれたグラフィックスとの組み合わせで、障害物の回避を行うことができる。

図3.2は、遠隔操作システムの具体的構成を示している。まず、作業環境の状況は、ロボット近傍に設置された作業領域全域を映し出すカメラとハンドロボットの先端効果器に取り付けられたクローズアップ用カメラの映像によって、人間オペレータに提示される。これらのカメラは各々2台ずつ用いられ、video signal switchによって切り替えられた映像をそのスイッチと連動するゴーグルを通して見ることによって、オペレータに立体視環

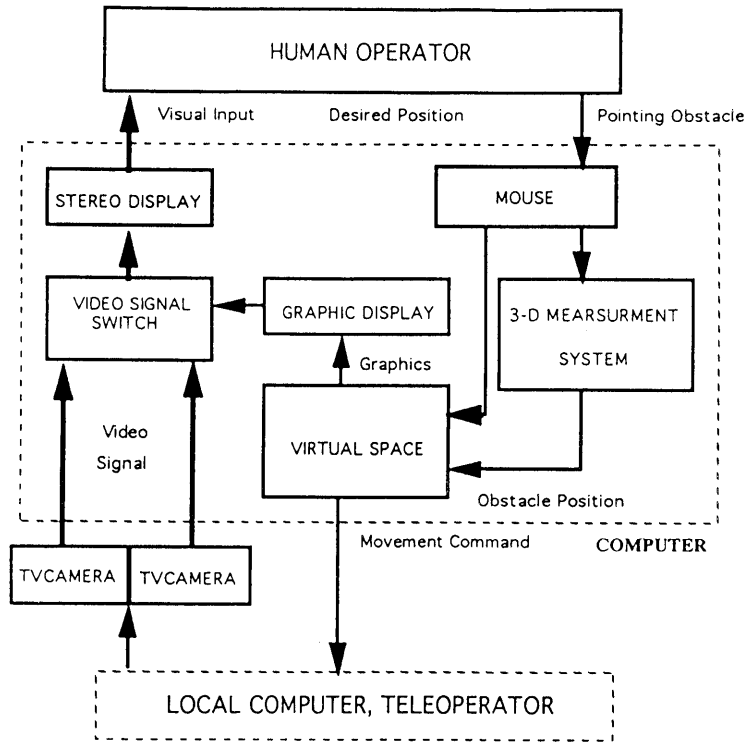


図 3.1 システムの概要

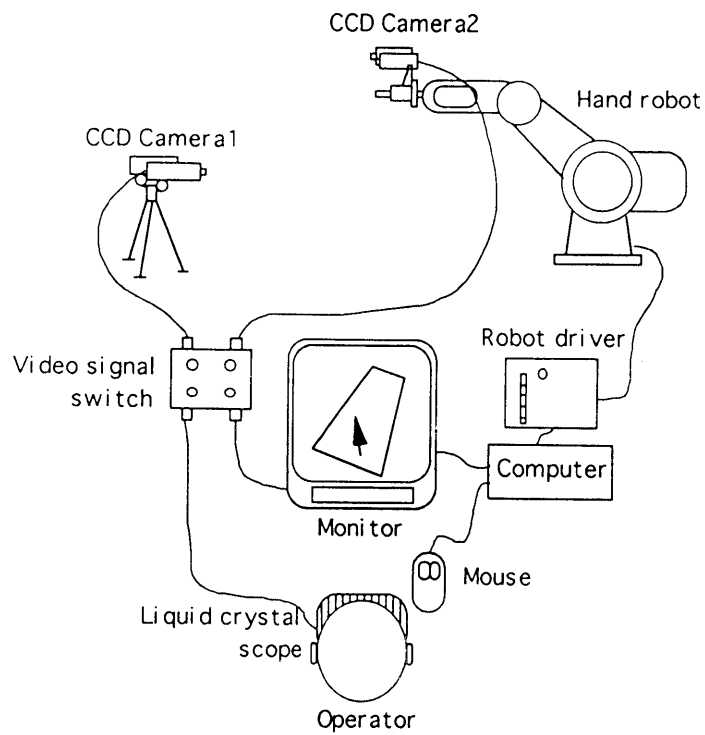


図 3.2 システムの構成

境が与えられる<sup>(26)</sup>。

立体視環境を与えるために、合計4台のカメラを用いているが、第2章で述べたようにこれらのカメラを用いて3次元の計測が可能である。2台のCCDカメラの映像に映っている対象物の任意の位置をマウスで指定することによって、その物体の3次元位置を計測することができる。

2台のカメラによる3次元計測においては、そのカメラの間の距離を広げることによって精度を向上させることができるが、本研究においては立体視のための映像と兼ねているために、カメラ間の距離を広げすぎると、立体視用映像として不適當となる。カメラの間隔を広げることができないと3次元情報の計測において精度の向上を期待することができないので、この対策としてロボットのハンド部にもカメラを設置し、3次元情報を2段階に分けて計測している。つまり固定カメラによって大域的情報をまず計測し、つづいて移動カメラによるクローズアップされた映像より精度の高い情報を得るようになっている。以上に述べた3次元計測システムによって、障害物の回避が不必要なときは、直接対象物の位置へ先端効果器を移動させることも可能である。

ここで、仮想空間と作業空間とのスーパーインポーズによるロボット制御について述べる。CCDカメラと作業空間との間の透視変換マトリクスを求めることによって、CCDカメラによって映し出される作業空間と仮想空間内に描いたグラフィックスとが空間的に対応がとれたものとなる。つまり透視変換マトリクスを用いて仮想平面を描き、作業空間とスーパーインポーズすれば、ロボットを制御する時に人間オペレータにとって大きな助けとなる。このシステムにおいては、ロボットの先端効果器の中央部を中心とした平面を描き人間オペレータに提示することによって、平面上の任意の点をマウスで指示するだけで、先端効果器を仮想平面上の指示された場所へ移動させることができる。

### 3.2.2 立体視システム

このシステムにおいて用いられている立体視システムは、時分割液晶眼鏡を用いた立体視を採用している。立体視システムの説明を図3.3に示す。2台のCCDカメラから送られてくるビデオ信号の一方（右眼用）の映像を偶数フィールドに、他方を（左眼用）の映像を奇数フィールドに表示するようなビデオ信号を発生し、ディスプレイに入力している。これにより、1/60秒毎に2台のCCDカメラの映像すなわち右眼用、左眼用の映像が交互にディスプレイに映るようになっている。それぞれの画像が対応する眼にだけ正しく映るためには、ビデオ信号の切り替えと同じタイミングで3Dスコープの右眼、左眼のシャッターを切り替える必要がある。右眼用ビデオ信号から同期信号処理回路によってフィールド判別信号を生成し、切り替え信号として使用した。また、2台のCCDカメラの同期を

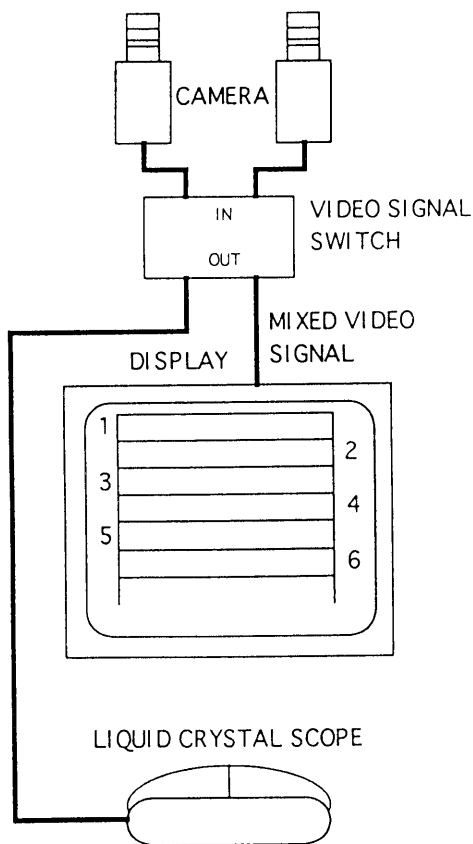


図3.3 時分割液晶眼鏡を用いた立体視システム

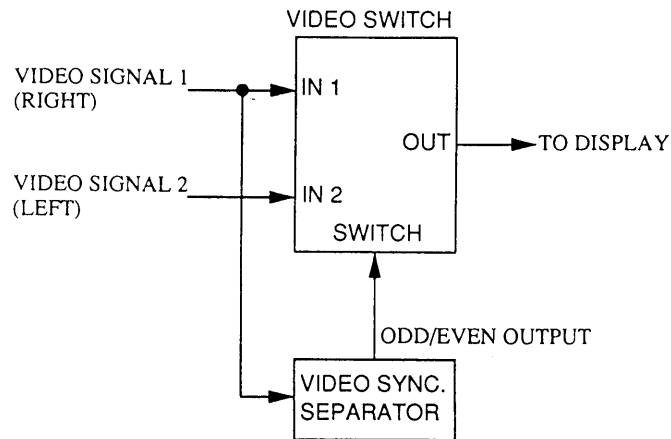


図 3.4 ビデオシグナルスイッチ

取るために、一方のカメラが出力するビデオ信号をもう一方の入力端子に入力している。これによって、完全なタイミングで映像を切り替えることができる。これらの一連の処理は、図 3.4 に示すビデオシグナルスイッチを用いて実現している。

この立体視システムは、ビデオ信号切り替えスイッチだけを必要とするだけで、立体視映像を家庭用ビデオデッキを用いて録画、再生することが可能で、非常に安価にシステムを構成することができる。

### 3.3 仮想平面を用いたロボット制御

#### 3.3.1 仮想平面の創成

3次元空間の中に仮想の平面を創成することによって、ロボットの制御に寄与することが出来る。例えば、2次元の入力装置であるマウスと仮想平面を組み合わせることによって3次元の入力装置として利用することが出来る。さらに、3次元空間の中を平面に沿っ

てカーソルを移動させることによってロボットのエンドイフェクターの姿勢制御も行うことが出来る。応用の詳細は後述する。

次のような方法で平面を創成する。まず、図3.5において $P_1$ ,  $P_2$ ,  $P_3$ を与えて長方形の頂点 $P_4$ ,  $P_5$ の座標を求める。

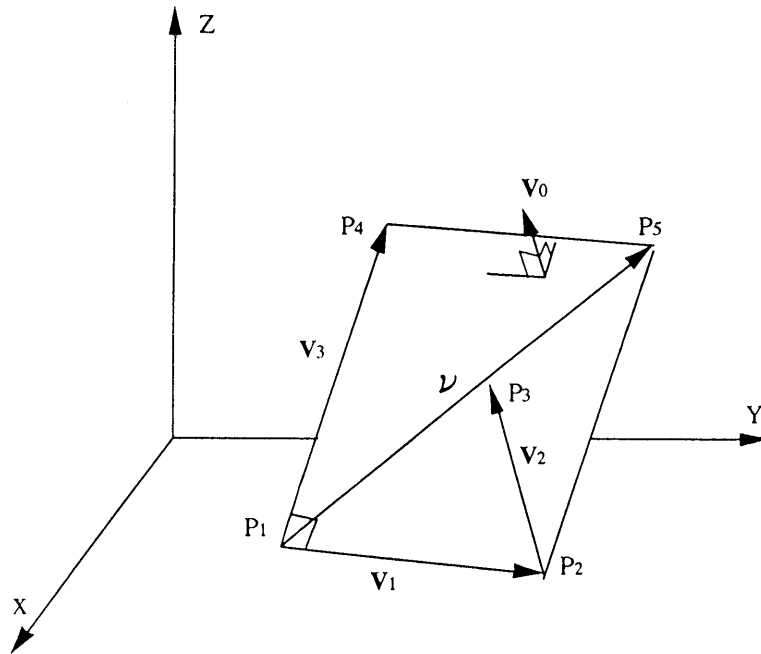


図3.5 仮想平面の創成

入力データを次のように与える。

$$P_1(x_1, y_1, z_1), P_2(x_2, y_2, z_2), P_3(x_3, y_3, z_3)$$

ベクトル $V_1$ ,  $V_2$ ,  $V_3$ の成分は、各々次のようになる。

$$V_1 \text{の成分} \quad a_1 = x_2 - x_1 \quad V_2 \text{の成分} \quad a_2 = x_3 - x_2$$

$$b_1 = y_2 - y_1 \quad b_2 = y_3 - y_2$$

$$c_1 = z_2 - z_1 \quad c_2 = z_3 - z_2$$

$$V_3 \text{の成分} \quad a_3 = x_4 - x_1 \quad V_0: \text{平面に垂直なベクトル}$$

$$b_3 = y_4 - y_1$$

$$c_3 = z_4 - z_1$$

3点  $P_1, P_2, P_3$  を通る平面を次式とすると、

$$a_0x + b_0y + c_0z + d = 0 \quad (3.1)$$

この面の法線ベクトル  $\mathbf{V}_0$  は

$$\mathbf{V}_0 = (a_0 \quad b_0 \quad c_0) \quad (3.2)$$

となる。次に  $\mathbf{V}_0$  と  $\mathbf{V}_3$  が直交、 $\mathbf{V}_1$  と  $\mathbf{V}_3$  が直交するから、

$\mathbf{V}_0 \perp \mathbf{V}_3$  より

$$a_0a_3 + b_0b_3 + c_0c_3 = 0 \quad (3.3)$$

$\mathbf{V}_1 \perp \mathbf{V}_3$  より

$$a_1a_3 + b_1b_3 + c_1c_3 = 0 \quad (3.4)$$

(3.3), (3.4) 式よりベクトル成分の大きさの比  $B_3, C_3$  を求める。但し  $A_3 = 1$  とする

$$B_3 = | (c_0a_1 - a_0c_1) / (c_1b_0 - c_0b_1) | \quad (3.5)$$

$$C_3 = | (b_0a_1 - a_0b_1) / (c_0b_1 - b_0c_1) | \quad (3.6)$$

ベクトル成分の比  $A_3, B_3, C_3$  が分かっている線分の長さ  $l$  が与えられたときのベクトル成分  $a_3, b_3, c_3$  を求める。

$$l = (a_3^2 + b_3^2 + c_3^2)^{1/2} \quad (3.7)$$

とおくと

$$a_3^2 + b_3^2 + c_3^2 = l^2 \quad (3.8)$$

となる。又、成分の比は等しいから

$$A_3/a_3 = B_3/b_3 = C_3/c_3 \quad (3.9)$$

$$b_3 = B_3 a_3 \quad (3.10)$$

$$c_3 = C_3 a_3 \quad (3.11)$$

となり、(3.8)式に代入すると、

$$a_3^2 + (B_3 a_3)^2 + (C_3 a_3)^2 = 1^2 \quad (3.12)$$

$$(1 + B_3^2 + C_3^2) a_3^2 = 1^2 \quad (3.13)$$

ベクトル成分  $a_3$ ,  $b_3$ ,  $c_3$  が次のように求められる。

$$a_3 = 1 (1 + B_3^2 + C_3^2)^{-1/2} \quad (3.14)$$

$$b_3 = B_3 a_3 \quad (3.15)$$

$$c_3 = C_3 a_3 \quad (3.16)$$

ベクトル成分  $a_3$ ,  $b_3$ ,  $c_3$  が分かれば  $P_4$  の座標  $(x_4, y_4, z_4)$  は

$$x_4 = x_1 + a_3 \quad (3.17)$$

$$y_4 = y_1 + b_3 \quad (3.18)$$

$$z_4 = y_1 + c_3 \quad (3.19)$$

次に  $P_5$  の座標  $(x_5, y_5, z_5)$  を求める。

$\nu = \mathbf{V}_1 + \mathbf{V}_3$  より

$$P_5 - P_1 = (P_2 - P_1) + (P_4 - P_1) \quad (3.20)$$

$$P_5 = P_2 + \mathbf{V}_3 \quad (3.21)$$

成分に分けると

$$x_5 = x_2 + a_3 \quad (3.22)$$

$$y_5 = y_2 + b_3 \quad (3.23)$$

$$z_5 = z_2 + c_3 \quad (3.24)$$

以上より、長方形の頂点の座標  $P_4(x_4, y_4, z_4)$ ,  $P_5(x_5, y_5, z_5)$  が求められる。

点  $P_1$ ,  $P_2$ ,  $P_4$ ,  $P_5$  の座標が求められると、次にこれらの3次元座標をディスプレイにグラフィックとして表示するため、透視変換マトリクスをかけて2次元座標に変換される。



### 3.3.2 仮想平面上の任意の点の2次元座標から3次元座標への変換

仮想平面を3次元空間の中に設定することによって、2次元入力装置であるマウスを使って3次元入力装置として利用することが出来ることを前節で述べたが、ここでは、仮想平面上の任意の点を与えることによって、その位置の3次元情報が得られることを示す。

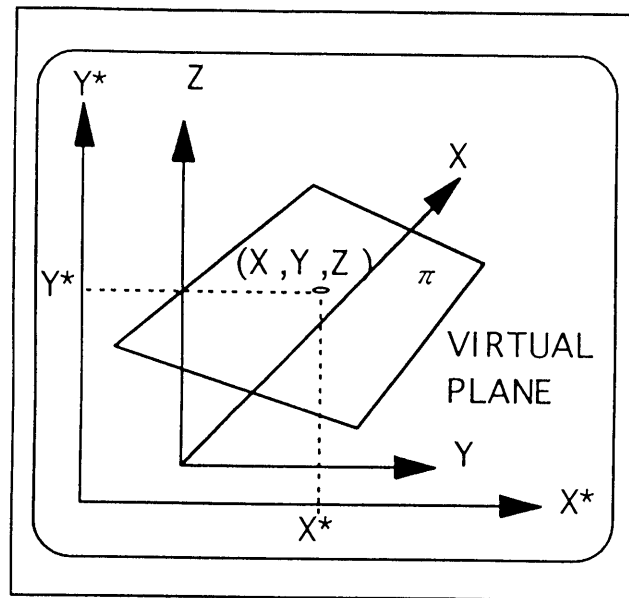


図3.6 オペレータに呈示される仮想平面

図3.6はオペレータに示されるディスプレイの様子を示したものである。図中の4角形がオペレータに呈示される仮想平面である。仮想平面上の任意の点をマウスカーソルで指示することによって、仮想平面上の任意の場所の3次元座標を求めることが出来る。

ディスプレイに表示された平面上の任意の点 $P(X^*, Y^*)$ から、3次元座標 $(X, Y, Z)$ を求める方法について述べる。

透視変換マトリクスの各要素が既知で、2次元映像の中のある点 $P(X^*, Y^*)$ が与えられると式(2.6), (2.7)が得られる。その式をもう一度以下に示す。

$$(T_{11}-T_{14}X^*)X+(T_{21}-T_{24}X^*)Y+(T_{31}-T_{34}X^*)Z+(T_{41}-T_{44}Y^*)=0 \quad (3.25)$$

$$(T_{12}-T_{14}Y^*)X+(T_{22}-T_{24}Y^*)Y+(T_{32}-T_{34}Y^*)Z+(T_{42}-T_{44}Y^*)=0 \quad (3.26)$$

式(3.25), (3.26)の係数を各々  $a_1, b_1, c_1, d_1, a_2, b_2, c_2, d_2$  と置くと

$$a_1X + b_1Y + c_1Z + d_1 = 0 \quad (3.27)$$

$$a_2X + b_2Y + c_2Z + d_2 = 0 \quad (3.28)$$

仮想平面  $\pi$  の式を

$$a_3X + b_3Y + c_3Z + d_3 = 0 \quad (3.29)$$

とすると、式(3.27), (3.28), (3.29)は  $X, Y, Z$  の連立方程式となるので、この式を解くことにより、 $X, Y, Z$  が求められる。

つまり、 $X^*, Y^*$  から  $X, Y, Z$  の3次元座標が求められる。

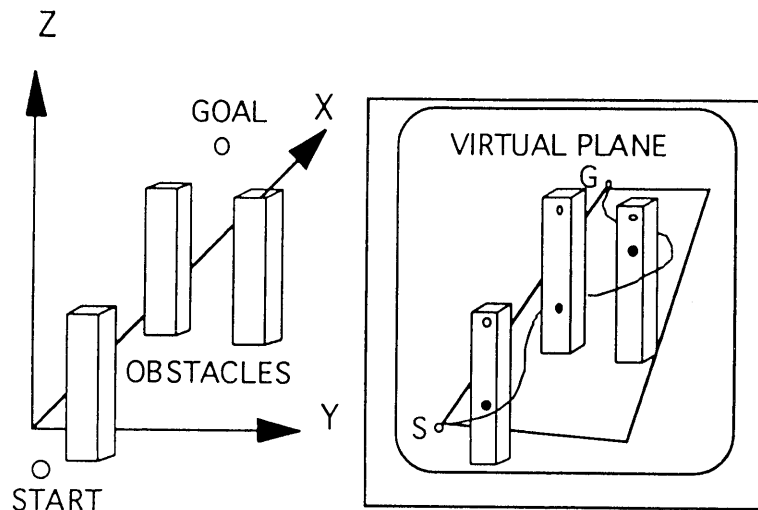


図3.7 障害物回避の一例

仮想平面を用いた障害物回避の応用例を、図3.7に示す。この図は、ロボットが働く作業環境の中に障害物がある場合の障害物の回避の例を説明している。本研究におけるシス

テムは第2章で述べたように3次元計測が可能であるので、図中の障害物の位置計測が可能であり、したがって仮想平面と障害物との交点を求めることが出来る。図中の黒丸がその位置を示している。

仮想平面上を移動する物体（ロボットのエンドイフェクターや、ヴィークル）が、障害物との衝突を回避するには、この黒丸の部分を回避することで可能となる。そこで、マウスカーソルを用いて、移動の行程を図のような曲線で指示することによって、物体を曲線に沿って移動させることが可能となる。つまり、マウスカーソルで入力された2次元座標が、上記の結果を用いて移動行程の3次元座標に変換されるからである。

### 3.3.3 マウスを用いた入力装置

従来は、6自由度のハンドロボットを制御する場合、ジョイスティックやマスターマニピュレータのような機械的入力装置が用いられてきた。前者を用いる場合、ハンドロボットの6自由度の移動方向とジョイスティックの操作方向との対応に習熟し、この装置を自由に操作するには多くの時間が必要であった。そのうえ監視カメラが移動して、視点が変化した場合、ロボットの移動方向とジョイスティックの操作方向が食い違うために誤操作を起こす恐れがあった。後者の場合は、オペレータの手の動きとロボットの動きの対応がとり易い利点があるが、微少距離移動や微少調整をする場合オペレータの手を安定させることが非常に困難となる欠点があった。

ここで提案している方法は次のような特長を持っている。

- (1) 入力装置としてマウスのみを用いるため、その他の機械的入力装置を全く必要としない。
- (2) ロボットハンドがどのような姿勢になっても、出力装置であるロボットと入力装置

の一部である仮想平面の座標系がいつも一致しているため制御が容易である。したがって仮想平面の上をマウスと連動するカーソルをあたかも面の上に絵を描くように移動させるだけで、ハンドロボットを制御することができる。

(3) カメラの移動による視点の変化に対しても、この方法では、仮想平面がカメラの移動とともに移動するだけで、オペレータの方向感覚には何等の影響も与えない。

(4) 微小距離移動や微小調整についても、オペレータのマウスを動かす手をテーブル上に固定することができるので、十分に対応が可能である<sup>(22)</sup>。

#### 3.3.4 仮想平面を使ったマウスによる6自由度制御

前節で、2次元入力装置であるマウスを使ってハンドロボットが制御出来ることを、そして、その手法の特長を述べた。この節ではその使用方法を具体的に説明する。すなわち、仮想平面とマウスと連動するカーソルの組み合わせによって、6自由度の制御を行う手法について述べる。

6自由度の制御とは、3次元空間の中で、任意の位置と任意の姿勢を指示することを意味するが、まず任意の位置への指示について述べる。

仮想平面上の任意の点をマウスカーソルで指示することによって、指示された点の3次元情報を得ることが出来る。さらに、仮想平面が任意の方向に回転できれば、作業空間中の任意の点を指示することが出来る。なお、仮想平面の回転については、3.3.6節で述べる。

図3.8は、仮想平面上での姿勢制御の方法を説明したものである。図中の矢印はエンドイフェクターが対象物を掴みに行くときの姿勢を表している。同図(a)は、点aにあるエンドイフェクターが前方の点bへ移動する場合を表している。

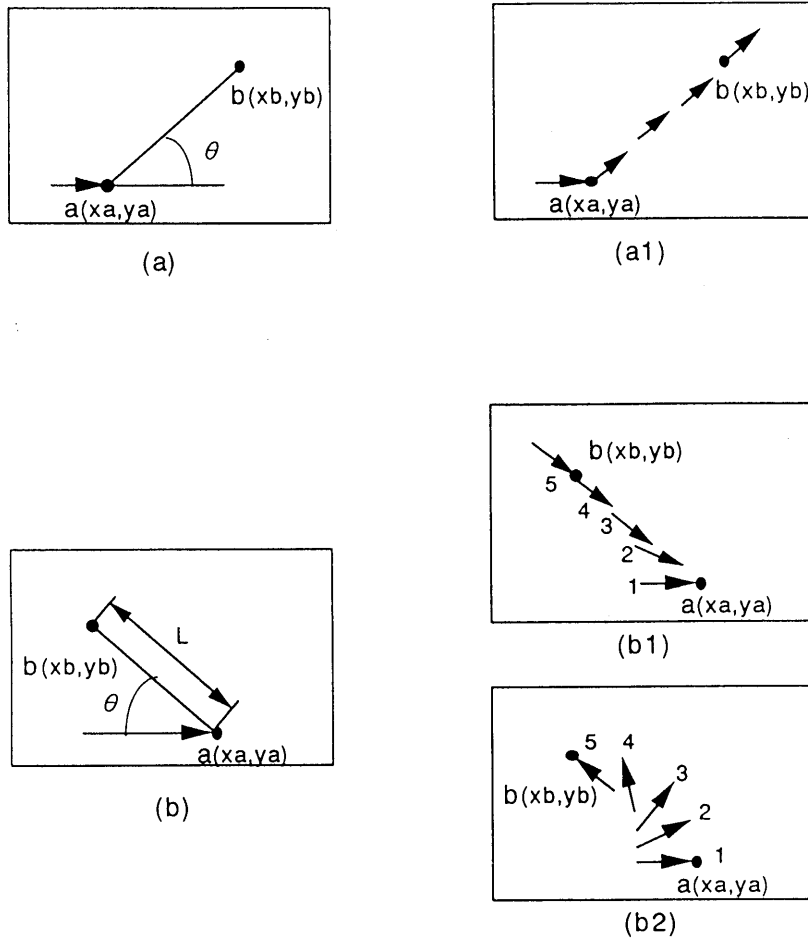


図 3.8 仮想平面上の姿勢制御

前方へ移動する場合は、同図 (a1) に示すように角度  $\theta$  を求め a 点で方向を角度  $\theta$  だけ回転し、点 b に向かって移動する。後方へ移動する場合は、同図 (b1), (b2) に示すように 2 通りの移動法が与えられる。この方法は、自動車を後方に移動させる場合に似せたものとなっている。つまり角度  $\theta$  と点 ab 間の距離  $L$  が小さいときは、その姿勢のままに後方へ移動する。反対に角度  $\theta$  と距離  $L$  が大きいときは、点 b に向かって徐々に方向転換をして移動する方式を採用している。本研究では、角度  $\theta$  が 45 度より小さく、距離  $L$  が画面上で 50 ドットより小さいとき図 3.8 (b1) の移動法を採用している。この場合も仮想平面が回転でき

るので、任意の姿勢を与えることができる。

### 3.3.5 ハンドロボットの制御

ハンドロボットの遠隔制御を行う際に、オペレータに呈示される立体視作業環境を図3.9に示す。この図はCCDカメラで撮影された作業環境内のハンドロボットの映像と仮想空間の中に描かれたエンドイフェクタの運動を制御するための仮想平面がスーパーインポーズされている様子を示している。仮想平面は、エンドイフェクタの移動用(a)とその平面の回転を制御するロボットのリスト部に描かれた平面(b)から構成されている。平面(a)の任意の点をカーソルでポインティングするとエンドイフェクタは指定された位置に移動することができ、平面(b)上の点を指示することによって平面(a)を任意の方向に傾斜させることができる。したがって、この2つの平面を用いることによってエ

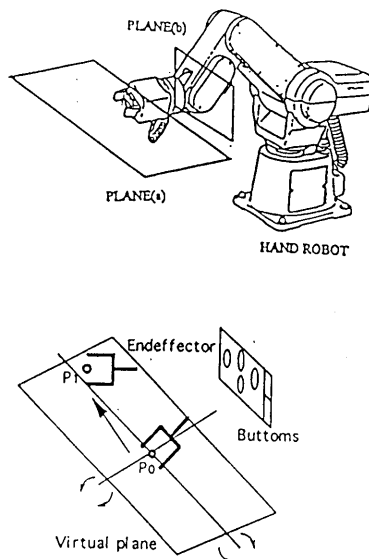


図3.9 オペレータに呈示される立体視作業環境

ンドイフェクターを任意の位置と姿勢に制御することができる。

さらに、エンドイフェクターのピッチ角も、先端効果器が移動する際に、自動的に変化するよう設計されている。ハンドロボットを制御する場合、ピッチ角を固定しておくこと、その移動可能範囲は非常に狭い範囲に限られてしまうので、このシステムにおいてはピッチ角の優先姿勢を最初に指定しておき、ある移動命令に対してその優先姿勢にできるだけ近い姿勢を保つように自動的に修正できるようになっている。この機能を付け加えたことにより、仮想平面上の移動点を指定するだけで、位置  $(x, y, z)$  とピッチ角  $p$  を制御することができるようになり、さらに、ハンドロボットの可動範囲を非常に広くとることができるようになった。

次に先端効果器のロール角とハンドの開閉について述べる。図 3.9 の平面 (b) において、ハンドロボットのリストの部分に描かれた円形のボタンは、ピッチ角、ロール角の制御およびハンド開閉のための制御ボタンである。このボタンをマウスでクリックすることによって各々の制御が可能となる。このボタンの表示位置は、ハンドロボットの移動に追従するようになっており、いつもリスト部の上に表示される。したがって、マウスのカーソルの移動量を最小限に抑えることができ、操作性の改善に貢献している。

### 3.3.6 仮想平面の回転

前節で仮想平面を用いることによって、二次元の入力装置を用いても、3次元の入力装置とし利用することが出来ることを示した。しかし、この場合は仮想平面が任意の方向に回転できることが前提となる。以下に仮想平面を回転させたときの処理について述べる。

図 3.10 のような平面  $\pi$  において、線分  $p_{\varepsilon}p_0$  を回転軸とする  $\theta$  回転と線分  $p_{\varepsilon}p_1$  を回転軸とする  $\phi$  回転について述べる。

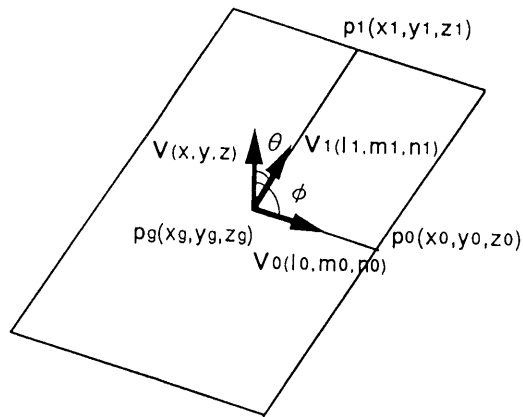


図 3.10 仮想平面の回転

まず  $\theta$  回転について述べることにする。平面  $\pi$  上の固定ベクトルを  $\mathbf{V}_0(l_0, m_0, n_0)$  とし、方向ベクトルを  $\mathbf{V}_1(l_1, m_1, n_1)$  更に  $\mathbf{V}_0$  のまわりに  $\mathbf{V}_1$  を角  $\theta$  回転したベクトルを  $\mathbf{V}$  とする。ここで  $|\mathbf{V}| = |\mathbf{V}_0| = |\mathbf{V}_1| = 1$  (単位ベクトル) とする。点  $P_g, P_0, P_1$  の座標を各々  $(x_g, y_g, z_g), (x_0, y_0, z_0), (x_1, y_1, z_1)$  とすると単位ベクトルの各成分は次の様に与えられる。いま、

$$l = x_0 - x_g, \quad m = y_0 - y_g, \quad n = z_0 - z_g \quad (3.30)$$

とおくと、 $\mathbf{V}_0$  の  $x, y, z$  成分は次のようになる。

$$l_0 = l / (l^2 + m^2 + n^2)^{1/2} \quad (3.31)$$

$$m_0 = m / (l^2 + m^2 + n^2)^{1/2} \quad (3.32)$$

$$n_0 = n / (l^2 + m^2 + n^2)^{1/2} \quad (3.33)$$

$$\text{また、} l = x_1 - x_g, \quad m = y_1 - y_g, \quad n = z_1 - z_g \quad (3.34)$$

とおくと、 $\mathbf{V}_1$  の  $x, y, z$  成分は次のようになる。

$$l_1 = l / (l^2 + m^2 + n^2)^{1/2} \quad (3.35)$$

$$m_1 = m / (l^2 + m^2 + n^2)^{1/2} \quad (3.36)$$



$$n_1 = n / (l^2 + m^2 + n^2)^{1/2} \quad (3.37)$$

つづいて、求めるベクトル  $\mathbf{V}$  の成分をそれぞれ  $(x, y, z)$  とする。

$$\mathbf{V}_0 \perp \mathbf{V}_1 \text{ より } l_0 l_1 + m_0 m_1 + n_0 n_1 = 0 \quad (3.38)$$

$$\mathbf{V}_0 \perp \mathbf{V} \text{ より } x l_0 + y m_0 + z n_0 = 0 \quad (3.39)$$

$\mathbf{V}_1$  と  $\mathbf{V}$  のなす角が  $\theta$  であるから

$$x l_1 + y m_1 + z n_1 = \cos \theta \quad (3.40)$$

ベクトル  $\mathbf{V}_0, \mathbf{V}_1, \mathbf{V}$  は各々単位ベクトルであるから

$$l_0^2 + m_0^2 + n_0^2 = 1 \quad (3.41)$$

$$l_1^2 + m_1^2 + n_1^2 = 1 \quad (3.42)$$

$$x^2 + y^2 + z^2 = 1 \quad (3.43)$$

となる。次に (3.38), (3.39), (3.42) より、 $x, y, z$  を求める。

$$\text{まず、式(3.38)より } l_0 x + m_0 y = -n_0 z \quad (3.44)$$

$$\text{式(3.39)より } l_1 x + m_1 y = -n_1 z + \cos \theta \quad (3.45)$$

となり、クラメールの定理より  $x, y$  は各々次の様に表される。

$$x = \frac{\begin{vmatrix} -n_0 z & m_0 \\ n_1 z + \cos \theta & m_1 \end{vmatrix}}{\begin{vmatrix} l_0 & m_0 \\ l_1 & m_1 \end{vmatrix}} = \frac{\begin{vmatrix} -n_0 & m_0 \\ -n_1 & m_1 \end{vmatrix}}{\begin{vmatrix} l_0 & m_0 \\ l_1 & m_1 \end{vmatrix}} z + \frac{\begin{vmatrix} 0 & m_0 \\ \cos \theta & m_1 \end{vmatrix}}{\begin{vmatrix} l_0 & m_0 \\ l_1 & m_1 \end{vmatrix}} \quad (3.45)$$

$$y = \frac{\begin{vmatrix} l_0 & -n_0 z \\ l_1 & -n_1 z + \cos \theta \end{vmatrix}}{\begin{vmatrix} l_0 & m_0 \\ l_1 & m_1 \end{vmatrix}} = \frac{\begin{vmatrix} l_0 & -n_0 \\ l_1 & -n_1 \end{vmatrix}}{\begin{vmatrix} l_0 & m_0 \\ l_1 & m_1 \end{vmatrix}} z + \frac{\begin{vmatrix} l_0 & 0 \\ l_1 & \cos \theta \end{vmatrix}}{\begin{vmatrix} l_0 & m_0 \\ l_1 & m_1 \end{vmatrix}} \quad (3.46)$$

式(3.45), (3.46)を次の様に変換する。

$$x = a_1 z + b_1 \quad (3.47)$$

$$y = c_1 z + d_1 \quad (3.48)$$

式(3.47), (3.48)を(3.42)に代入すると

$$(a_1 z + b_1)^2 + (c_1 z + d_1)^2 + z^2 = 1$$

$$(a_1^2 + c_1^2 + 1) z^2 + 2(a_1 b_1 + c_1 d_1) z + b_1^2 + d_1^2 - 1 = 0 \quad (3.49)$$

式(3.49)を  $e_1 z^2 + f_1 z + g_1 = 0$  とおき、この2次方程式を解くと  $Z$  が得られる。

$$z = (-f_1 \pm (f_1^2 - 4 e_1 g_1)^{1/2}) / (2 e_1) \quad (3.50)$$

つづいて  $x, y$  は式(3.47), (3.48)に  $Z$  を代入することにより次の様に求められる。

$$x = a_1 z + b_1$$

$$y = c_1 z + d_1$$

ここで求められた  $x, y, z$  は単位ベクトルの要素であるから仮想平面  $\pi$  が角  $\theta$  回転した時の点  $p_1$  の座標を求める必要がある。新しい点  $p_1$  の座標を  $(x_1, y_1, z_1)$  とすると

$$x_1 = x \left( (x_1 - x_\theta)^2 + (y_1 - y_\theta)^2 + (z_1 - z_\theta)^2 \right)^{1/2} + x_\theta \quad (3.51)$$

$$y_1 = y \left( (x_1 - x_\theta)^2 + (y_1 - y_\theta)^2 + (z_1 - z_\theta)^2 \right)^{1/2} + y_\theta \quad (3.52)$$

$$z_1 = z \left( (x_1 - x_\theta)^2 + (y_1 - y_\theta)^2 + (z_1 - z_\theta)^2 \right)^{1/2} + z_\theta \quad (3.53)$$

となり、平面  $\pi$  が  $\theta$  回転した時の新しい点  $p_1$  の座標が求まると矩形の角の座標 ①②③④

は次の様に求められる。その様子を図3.11に示す。

点 ④ :  $\mathbf{V}_1 - \mathbf{V}_0$  より

$$\left( (x_1 - x_\theta) - (x_0 - x_\theta) + x_\theta, (y_1 - y_\theta) - (y_0 - y_\theta) + y_\theta, (z_1 - z_\theta) - (z_0 - z_\theta) + z_\theta \right) \quad (3.54)$$

点 ③ :  $\mathbf{V}_1 - \mathbf{V}_0$  より

$$\left( (x_1 - x_\theta) + (x_0 - x_\theta) + x_\theta, (y_1 - y_\theta) + (y_0 - y_\theta) + y_\theta, (z_1 - z_\theta) + (z_0 - z_\theta) + z_\theta \right) \quad (3.55)$$

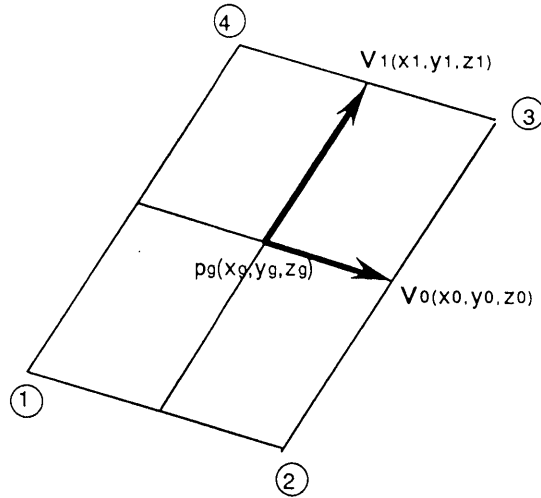


図 3.11 仮想平面の拡張

点 ② :  $\mathbf{V}_0 - \mathbf{V}_1$  より

$$( (x_0 - x_g) - (x_1 - x_g) + x_g, (y_0 - y_g) - (y_1 - y_g) + y_g, (z_0 - z_g) - (z_1 - z_g) + z_g ) \quad (3.56)$$

点 ① :  $-\mathbf{V}_1 - \mathbf{V}_0$  より

$$( -(x_1 - x_g) - (x_0 - x_g) + x_g, -(y_1 - y_g) - (y_0 - y_g) + y_g, -(z_1 - z_g) - (z_0 - z_g) + z_g ) \quad (3.57)$$

### 3.4 結 言

人間に立体視環境を呈示できる遠隔操作システムの構築と、仮想平面を用いたロボットの制御について述べた。

仮想平面を用いたロボット制御においては、システムへの入力状況を仮想空間の中に操作空間として表示した。操作空間は、仮想平面とマウスの組み合わせによって構成されている。マウスは2次元の入力装置であるが、空間の中で任意の方向に回転出来る仮想平面を設置することによって、6自由度の制御が可能となった。

さらに、ロボットが作業をする実空間と、操作空間としての仮想空間をディスプレイ上

にスーパーインポーズして表示することによって、インタフェースの改善を図った。このことにより、人間の感覚にマッチした形態で入力と出力の両者の情報を提示することが可能となった。例えば、作業空間の中のハンドロボットをある位置へ移動させる場合、その位置にマウスを移動させて指示するだけで、ハンドロボットは所望の位置に移動することが出来る。また、そのとき立体視環境は、人間の位置認識の能力を高めることに貢献している。

## 第4章 仮想平面を用いた衝突回避と3次元空間の位置認識

### 4.1 緒言

この章では、仮想平面を用いた障害物の回避問題と立体視環境のもとでの3次元空間の位置認識について述べる<sup>(9) (15) (16)</sup>。

前章では、仮想平面を用いたハンドロボットの制御について述べたが、この章では、仮想平面を使った障害物回避について述べる。ここでは2つの回避法を提案している。最初は、エンドイフェクターが物体を把持している状態で、障害物を回避しようとするものであり、概略的な障害物回避経路はオペレーターによって入力される。その後は経路をコンピュータが逐次修正を加え、完全な回避経路を得ようとするものである。次に、仮想平面上に描かれた曲線の追跡を利用した障害物回避について述べている。この方法は、あたかも人間が図面上に置かれた定規に沿って鉛筆で曲線を引くように、平面上に描かれた曲線を追跡しながら、ハンドロボットのエンドイフェクターの制御を行うものである。最後に、立体視環境のもとでの3次元空間の位置認識について述べる。

### 4.2 経路修正法による障害物回避

#### 4.2.1 対象物体を掴んだエンドイフェクターの障害物回避

ハンドロボットの作業として、作業環境へ物体を運び入れたり、逆に物体を作業環境の外へ運び出すことが考えられる。ここでは、エンドイフェクターが物体を把持している状態で、障害物を回避する問題を考えることにする。

まず前提となるのは、第3章で述べたように3次元空間に設定された仮想区平面とそしてマウスと連動するカーソルによって、エンドイフェクターの制御が可能であることである。

図4.1は、ここで想定している環境を説明したものである。エンドイフェクターが現在いる場所をスタート地点として、何らかの物体を障害物を回避しながら、ゴールに移動させる状況を考える。第2章で述べたように、このシステムは、3次元計測システムを搭載しているので、ゴールや障害物の位置は計測可能であることを前提としている。また、エンドイフェクターが運ぶ物体は、問題を単純化するため直方体であると仮定している。このシステムでのオペレータの仕事は、スタート地点からゴールへの概略的な衝突回避経路を図4.1の破線のように指定するだけである。なお、図中の平面は、スタート点とゴール点を含む仮想平面を表している。

図4.2は、エンドイフェクターが把持している物体の様子を説明している。点 $P^{\circ}$ は物体の中央部、点 $P^{\circ}$ は物体の前面部を意味している。オペレータが、マウスを用いて点 $P^{\circ}$ の移動先を順次指定することによって、物体は並進または回転をしながら移動することが出来る。この様にして、物体の移動経路がオペレータによって入力されると、後はコンピュータがその経路を修正しながら、衝突を回避できる経路を自動的に定めて行く。

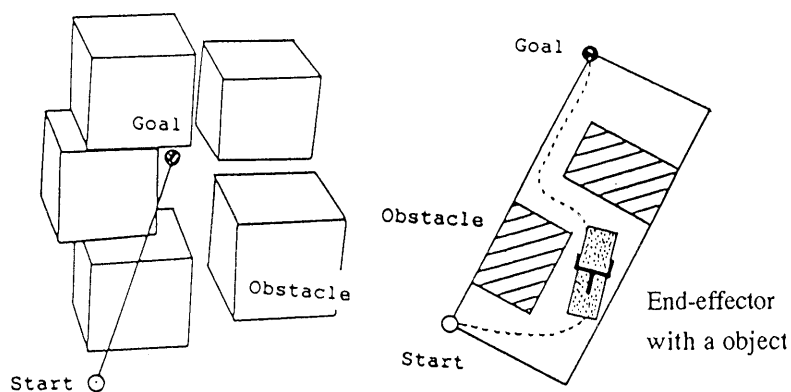


図4.1 物体を把持したエンドイフェクターの制御

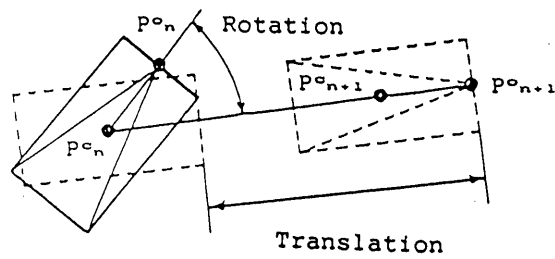


図 4.2 物体の移動を制御する方法

#### 4.2.2 物体と障害物の接触条件と経路の修正

オペレータが入力する回避経路は、完全なものである必要はない。なるべく障害物と衝突しない経路が選ばれれば、修正のために必要な時間は節約できるが、その場合は、入力に時間がかかることになる。オペレータによる概略的経路の入力後は、その経路を修正し

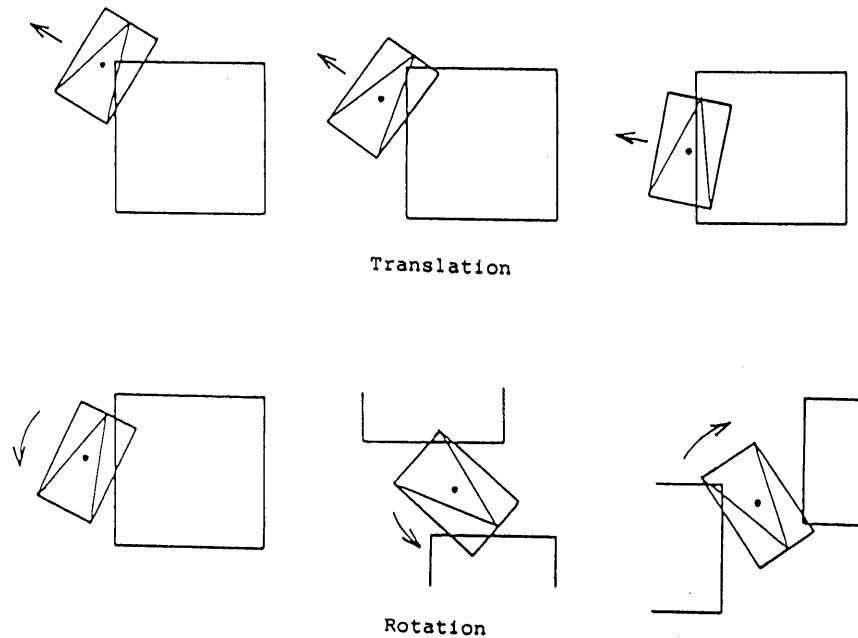
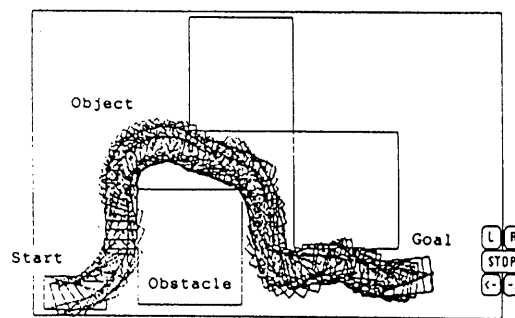


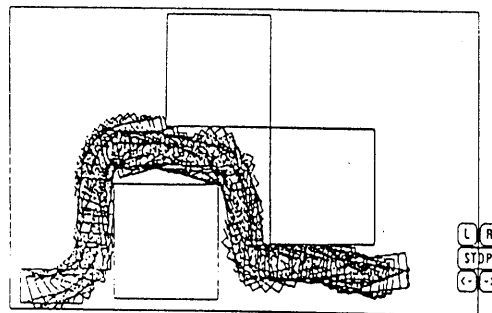
図 4.3 接触条件と修正法

て、障害物との衝突を完全に回避してゴールへ移動する経路を求めなければならない。図4.3は、物体と障害物との接触条件を示している。図中の矢印は、衝突回避の運動方向を示しており、合計6種類の並進回避と回転回避を考えている。つまり図のような接触をしている時は、各々の矢印の方向に回避行動をするようにプログラムされている。

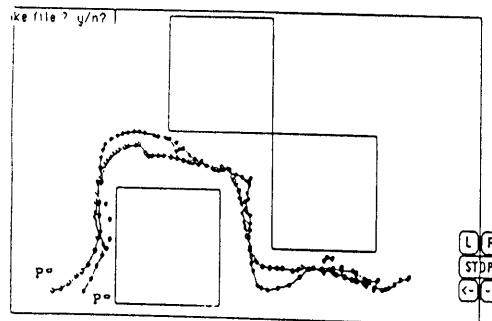
図4.4は、経路修正の結果の一例を示したものである。(a)はオペレータによって入力された概略的経路である。所々で、障害物と衝突しているのが分かる。(b)は修正された後



(a) Outline path selected by operator



(b) Modified path without collision



(c) Locus of points  $P^c$  and  $P^o$  after modification

図4.4 経路修正の一例



の物体の移動経路である。巧みに衝突を回避していることが分かる。(c)は、修正後の物体上の点  $P^*$  と  $P^*$  の移動軌跡である。物体の移動に際して、これらの求められた点  $P^*$  と点  $P^*$  を与えれば、障害物を回避できることになる。

#### 4.2.3 考察

障害物が存在する作業環境の中で、衝突を回避するための方法として、オペレータが選んだ概略的障害物回避経路を仮想物体を用いて逐次修正する方法を提案し、その有効性をシミュレーションを用いて確かめることが出来た。このシステムにおいて、もし回避経路が求められない場合は、再度オペレータが概略的回避経路を入力することによって解決出来る。

#### 4.3 仮想平面上に描かれた曲線を用いた障害物回避

ロボットをある任意の位置まで移動させたい時、途中に障害物がある場合には、それを回避して移動させてやらなければならない。第3章3.3.2節でその一例を示した。その方法は、マウскарソルを用いて平面上の移動点を入力し、逐次その2次元座標を3次元座標に変換して、移動に必要な3次元座標を求めたものであった。この方法は、二次元入力装置であるマウスを使って6自由度を制御する場合に用いられるもので、制御の詳細は第3章3.3.4節で、仮想平面を使ったマウスによる6自由度制御ですでに述べられている。

ここで提案している方法は、仮想平面上に描かれた曲線をグラフィック情報として用いることに特長がある。

ここでは、図4.5に示される様な障害物回避について述べることにする。まず障害物の

近くに図のように障害物を回避出来るようなルートを仮想平面上の曲線として描く。続いて、曲線を構成する点群の検索を行い、エンドイフェクターが移動する3次元空間の座標を逐次計算してロボットをコントロールするものである。

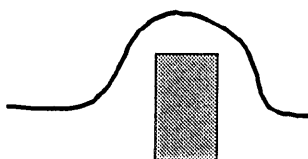
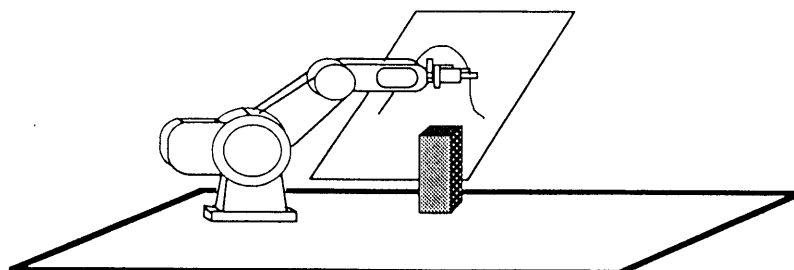


図4.5 障害物の回避

#### 4.3.1 仮想平面上に描かれた曲線の追跡

仮想平面上の任意の点は、ディスプレイ上の平面の座標  $(x, y)$  が求められれば、3次元座標に復元することが出来る。そこで、ディスプレイ上に描かれた曲線を追跡する手法について考える。図4.6の斜線部はディスプレイに描かれた曲線の一部を表している。まず、曲線の上端部を検索のスタート地点と仮定する。スタート地点の回りをgetpixel関数（C言語の関数）を用いて、周りの色を調べる。検索の順番は図4.6のように左上より時計回りとし、1番から8番までの8点とする。指定した色つまり曲線を構成する色が見つ

かったらそこで検索を止め、その地点に移動する。この場合は指定した色が4番目に見つ
 かったので、それ以降の検索はせずに4番目の位置である右に移動する。ここでミスを防
 ぐために一度通った点は putpixel関数 (C言語関数) で違う色に塗り直すようにしている。
 そして移動した点でも先程と同じようにその回りを左上より時計回りに検索していく。こ
 の場合はまた右に行くことになる。このように検索と移動を何度か繰り返すことによって
 ディスプレイ上に描かれた曲線を追跡することが出来る。そして曲線の下端の最後の点ま
 で行って、もう周りに指定した色が無ければその位置で止まり検索を終了する。

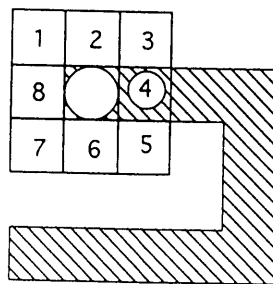


図 4.6 曲線の追跡

#### 4.3.2 曲線追跡シミュレーション

前節で提案した方法を用いて、まずパソコンのディスプレイ上でシミュレーションを行
 った。図 4.6 に示すようにロボット及び仮想の平面は実世界と対応されるために、あらか
 じめ測定された透視変換マトリクスを用いてディスプレイに表示される。よってこの空間
 はロボットが作業する実世界と対応している。このディスプレイ上の仮想の平面上にマウ
 スを使って障害物を回避出来るような曲線を描く。検索のプログラムにより曲線の2次元
 座標  $(x, y)$  を求め、その値を3次元座標  $(x, y, z)$  に変換する。この値をロボットのエンドイ
 フェクターの先端座標としてロボットをコントロールする。これによってロボットが曲線

に沿って仮想平面上を移動し、障害物を回避する。

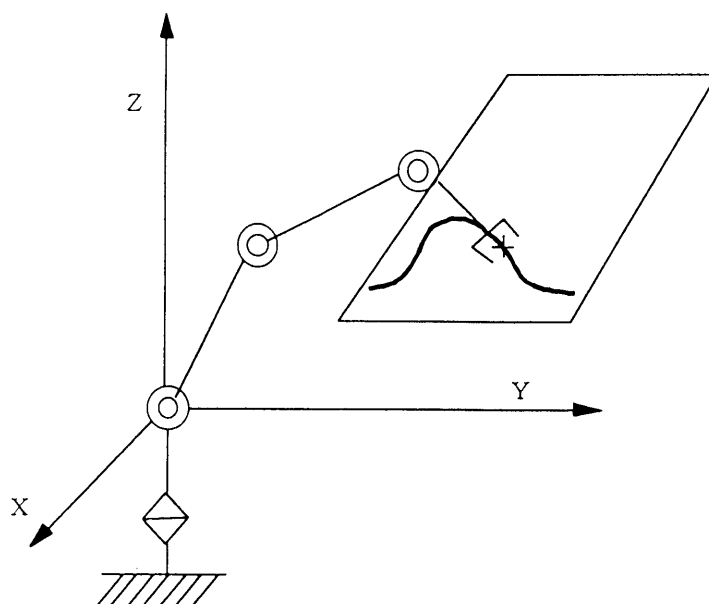


図4.7 障害物回避のシミュレーション

#### 4.3.3 実ロボットを使つての障害物回避

前節で述べたパソコンでのシミュレーションを、今度は実ロボットを使つて回避実験を行った。ロボット及び障害物の映像を2台のカメラで取り込み、ディスプレイ上に表示する。つづいて、この映像と仮想平面とをスーパーインポーズにより重ねる。そして最後に障害物の位置を確かめながら仮想平面上に曲線を描いていく。このシステムでは、仮想平面上に描かれた曲線を追跡し、その2次元座標を3次元座標に変換して、ハンドロボットのエンドイフェクターを良好に制御することができた。

#### 4.3.4 考察

回避のための曲線が描かれれば、後はこの曲線をグラフィック的に追跡し、逐次2次元座標を3次元座標に変換して、障害物の回避が可能となる。この新しい提案によって、紙に書かれた曲線が、あたかも曲線定規のように用いられことが確かめられた。つまり人間が、紙の上に直線や曲線を描く時、定規を用いて、鉛筆の動きを制御出来るように、仮想平面上に描かれた曲線が、あたかも定規のように、ロボットのエンドイフェクターの動きを制御出来ることを確かめることが出来た。

#### 4.4 立体視環境における3次元空間の位置認識

従来は、2次元ディスプレイに映し出された作業環境の映像をもとにロボットの遠隔操作が行われてきた。しかし、この場合、奥行き方向の情報の欠落のためにロボットの操作は非常に困難であった。作業環境の中で、床に相当する水平面を環境の中で見つけることができる場合は、ロボットや対象物の床の上に置ける位置関係から辛うじて奥行き方向を認識してきた。しかし、呈示された映像の中で床や机のような水平面を見つけない場合は、人間オペレータに非常に大きな負担をかけてきた。

本研究においては、時分割液晶眼鏡を使用して、オペレータに立体視環境を提供することが出来るようになった。そこで、立体視システムを用いて、ロボットの遠隔操作を行うとき、人間の位置認識にどの程度の改善がなされるかについて実験的に検討した。

##### 4.4.1 立体視環境における3次元空間の位置認識の実験

人間に3次元立体視環境を与えたとき、従来の2次元表示に比べどの程度位置認識が改

善されるかを、表4.1に示す実験を行って検討した。実験は大きく分けて4つに分類される。

実験#1：実世界の中に置かれた目標物と実物体のカーソルが2次元ディスプレイに映されているときの位置認識。

実験#2：実験#1の実験環境が立体視の環境で呈示される場合の位置認識。

実験#3：カーソルが実物体の代わりに、仮想空間に描いた仮想カーソルを用いた場合の位置認識（目標物は実物体）。

実験#4：目標物とカーソルの両者ともに仮想空間に描かれた場合の位置認識。

表4.1 3次元位置認識の実験

実験	目標物体	カーソル	呈示法
#1	実物体	実物体	2D
#2	実物体	実物体	3D
#3	実物体	仮想カーソル	3D
#4	仮想物体	仮想カーソル	3D

被験者に呈示される実験用環境を図4.7に示す。図示されている環境をCCDカメラからの映像としてディスプレイに表示される。なお、CCDカメラと作業環境の原点との距離は2000mmである。被験者として、20才の健康な男子3名が用いられている。ターゲットは直径15mm、高さ185mmの円柱であり、実物体のカーソルは高さ15mm、直径8mmの円柱である。仮想カーソルは、三角垂の頂点部の形状をしており、1辺が16mmである。実験#3と実験#4においては、仮想カーソルを被験者に立体視環境のもとで呈示する

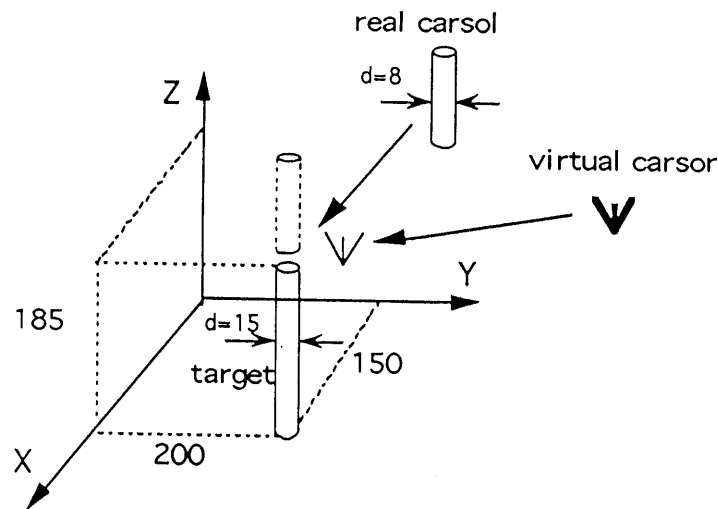


図4.8 位置認識の実験環境[単位 mm]

が、従来の方法では、画面切り替え時に、左目用に呈示される画像が、両目に一瞬ではあるが呈示されることがあって、被験者にとって安定して立体像を想起させる妨げになっていた。そこで、次節で述べるような改善を図った。

#### 4.4.2 カーソルの立体表示の改善

まず、本研究で用いている立体視のための出力図形の作り方について、図4.8を用いて説明する。立体視を行うには右目用の画像(a)と左目用の画像(b)を制作する。つづいて、左目用画像を奇数ラインだけの映像(c)に、そして右目用画像を偶数ラインだけの映像(d)にそれぞれ加工する。最後に2つの画像を同じ画面上に描き(e)人間に呈示する。ディスプレイには、画像(e)が表示されるが、液晶シャッター付きのスクープを用いることによって、人間の左目には画像(c)が、右目には画像(d)が呈示されることになり、立体視を実現している。

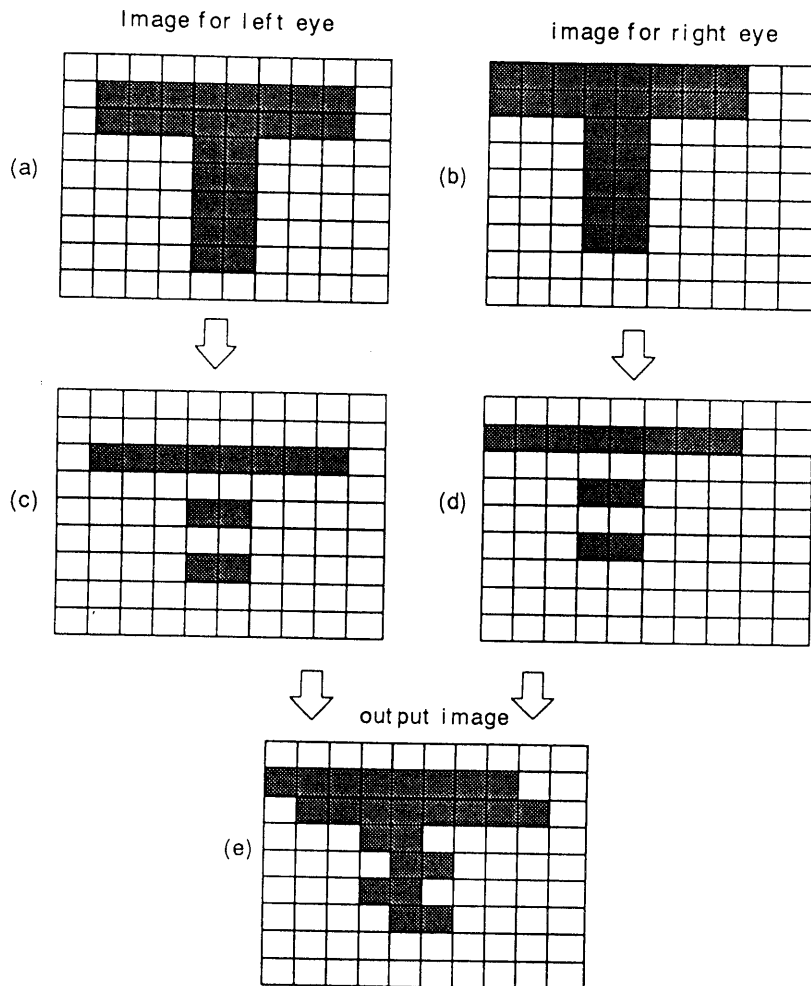


図4.9 立体視のための図形表示法

しかし、従来の方法では先に述べた欠点が見られたので改良を加えることとした。まず従来の方法について図4.9を用いて説明する。表画面にまず左目用の画像(a)を描く。その後、裏画面に右目用の画像(b)を描いてから、表画面のY座標の偶数ラインに裏画面の偶数ライン上の映像(d)のみを出力し、立体視用の画面(e)を形成していた。しかし、この方法では、裏画面に画像を描いている時間と、表画面に裏画面をコピーする時間は、左目用の画面(c)全体が一度表画面に描き出されているために、右目に不要なラインが見えることになる。そのため、重ね合された画面がちらつき、目が疲れたり、立体に見えにくくなる



状況が生じた。

そこで、図4.10に示すように、画像を描く時点で処理をするため、左目用の画面を描く場合には奇数ラインのみに、右目用の画面を描く場合には偶数ラインのみに描くための描画関数を制作した。このような改善によって、画面のちらつきを大幅に押さえることができた。また、この描画関数を用いることで、立体視用画像を容易に描くことが可能となった。

さらに、次のような改善も同時に行われた。従来の方では、左右の目に呈示する画像を作成するのに、表画面と裏画面の2つの画面が必要であったが、改善後は、1画面で処

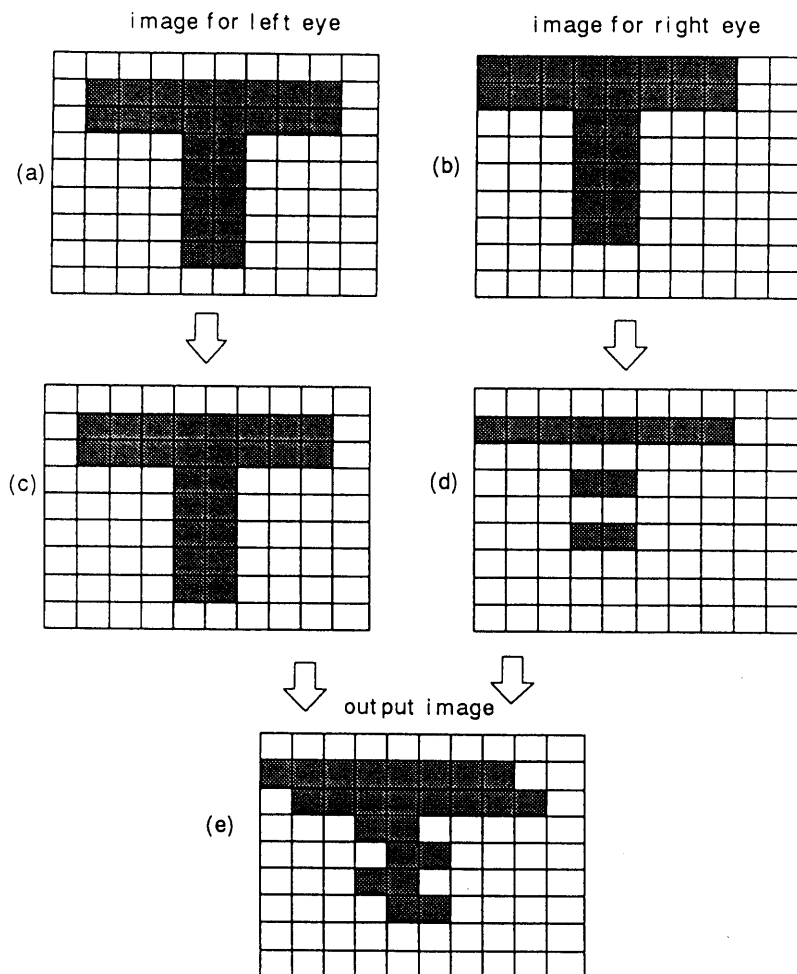


図4.10 立体視用図形表示法（従来方式）

理出来るようになった。そのため画像を更新する場合は、現画像を人間に呈示している間に、更新用の画像を別の裏画面に描き、完了と同時に更新画像に切り替える方式を採用した。その結果、2画面を使ってスムーズな動きを呈示させることが可能となり、従来よりも快適な立体視の環境を実現した。

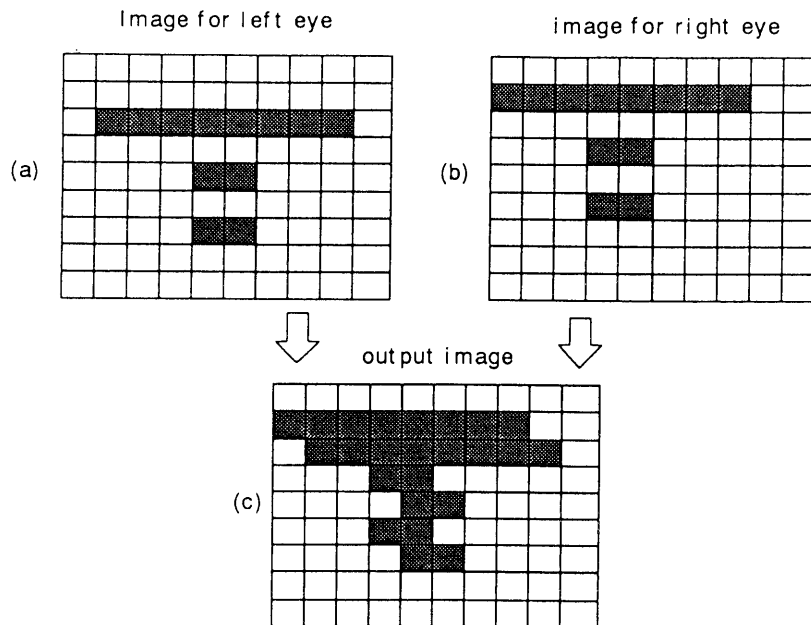


図 4. 1 1 立体視用図形表示法（改良型）

#### 4.4.3 位置認識実験の結果（実験条件#1、#2、#3）

表 4. 2 は、表 4. 1 で説明した実験の結果を示している。表中の数値は、位置認識の誤差を表している。つまり、呈示されているターゲットの位置へカーソルを移動させて、被験者が両者が一致したと認識したときの両者間の距離をmmで表している。数値が小さいほど、良好な位置認識をしていることを表している。なお、実物体の目標物体は直径 15 mm、カーソルは直径 8 mm の円柱を用いている。

表 4.2 位置認識実験#1、#2、#3の結果

No.	#1(mm)			#2(mm)			#3(mm)		
	sub1	sub2	sub3	sub1	sub2	sub3	sub1	sub2	sub3
1	17	220	5	5	37	30	14	32	80
2	255	120	350	20	50	0	16	16	30
3	130	130	400	20	0	15	85	77	31
4	125	60	125	30	0	0	51	22	42
5	45	145	17	36	3	40	28	21	53
6				10	3	40	46	30	19
7				35	30	5	51	22	52
8				50	25	0	15	108	36
平均	114	135	179	26	19	16	39	41	43

実験#1は、従来の2Dディスプレイで、CCDカメラから送られてくる映像を観測するときの位置認識の結果であり、実験#2は、立体視環境の中での位置認識の結果である。実験#1では、ほとんど位置認識が出来ないことを示している。しかし実験#2では、平均的に20mm前後まで位置認識が可能であることを示しており、立体視環境の効果が顕著に現れている。

実験#3は、実物体のターゲットと仮想カーソルとの間の位置認識の結果である。この結果は、前述の実験#2の実物体同士の位置認識誤差の約2倍程度の誤差となっている。これは、カーソルを実物体から仮想カーソルに変えた影響を示すもので、人間に呈示するグラフィックカーソルの呈示法に改善が必要であることを示している。

#### 4.4.4 位置認識実験#4の実験方法

実験は、コンピュータのディスプレイ上に、2つの透視変換マトリクスを用いて2つの図を描き、立体視スコープを通してディスプレイを見ることにより、被験者に立体視環境

を与えて行われる。図4.12は、被験者に与えられる立体視環境であり、立方体の頂点にある円は位置認識を実験するためのターゲットである。被験者に与えられる立方体の寸法は次の通りである。

$$z = 30 \text{ cm} \quad x = 40 \text{ cm} \quad y = 38 \text{ cm}$$

なお、表示の大きさは、この立方体の頂点がすべてディスプレイに表示される範囲内で、最大の表示となるようにした。まず、ディスプレイ上に立方体の8個の頂点のうちの1つがターゲットとして表示される。被験者には、このターゲットの位置に仮想のカーソルを移動させ、一致させるように求められている。

カーソルの移動にはテンキーが用いられている。奥行き方向zは、1と9を、横方向xは、4と6を、縦方向yは、8と2のキーをそれぞれ押すことによって、仮想カーソルは移動する。

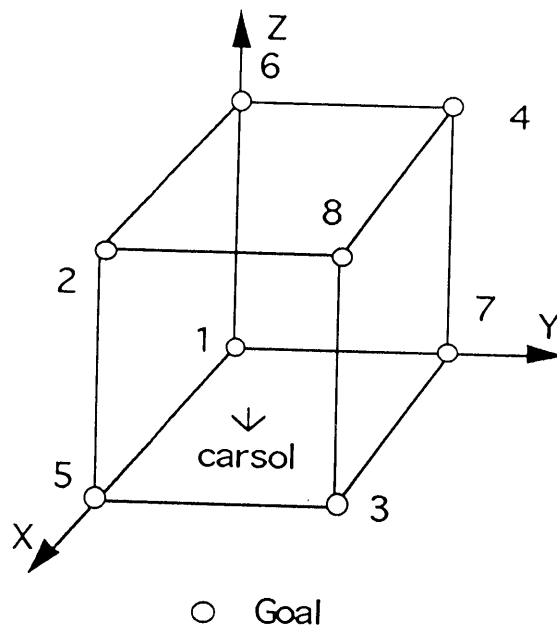


図4.12 位置認識実験の立体視環境

#### 4.4.5 実験結果

実験は次のように行われた。立方体の頂点8点の内一点が表示され、表示された目標位置の所へカーソルを移動させ、一致したと被験者が判断した時、指定されたキーを押すことになっている。表示される点の順番はランダムに設定されており、キーを押すことによって、順次8点が表示され実験が進行していく。この一連の実験を5回繰り返すことになっている。この実験では被験者は3名を用いている。ある程度操作に慣れさせるために、8点の位置認識実験を2回行っている。実験結果を表4.3に示す。

表4.3 実験#4の結果

試行 物体	sub1					sub2					sub3				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	0	0	51	51	-	37	233	92	0	0	59	0	0	102	51
2	0	0	14	14	-	79	14	617	0	14	252	51	0	14	14
3	57	33	70	46	-	24	82	211	70	24	124	79	42	0	0
4	22	0	0	0	-	0	22	64	0	36	23	46	32	0	0
5	0	0	0	30	-	104	51	23	14	60	67	67	62	14	0
6	37	0	14	0	-	36	36	45	0	0	73	109	90	50	87
7	24	136	0	37	-	49	46	66	0	0	176	135	70	0	10
8	22	220	45	22	-	22	22	0	45	22	167	145	64	0	46
平均	20	49	24	25	-	44	63	152	16	20	144	79	45	23	26

#### 4.4.6 考察

図4.13より、試行回数が増えるに従って誤差が減少していくことが分かる。Sub.1については、試行回数1の時から非常に誤差が小さい結果となったので、試行回数4で実験を打ち切っている。Sub.2の試行回数3の時非常に大きな誤差が現れているが、この原因は

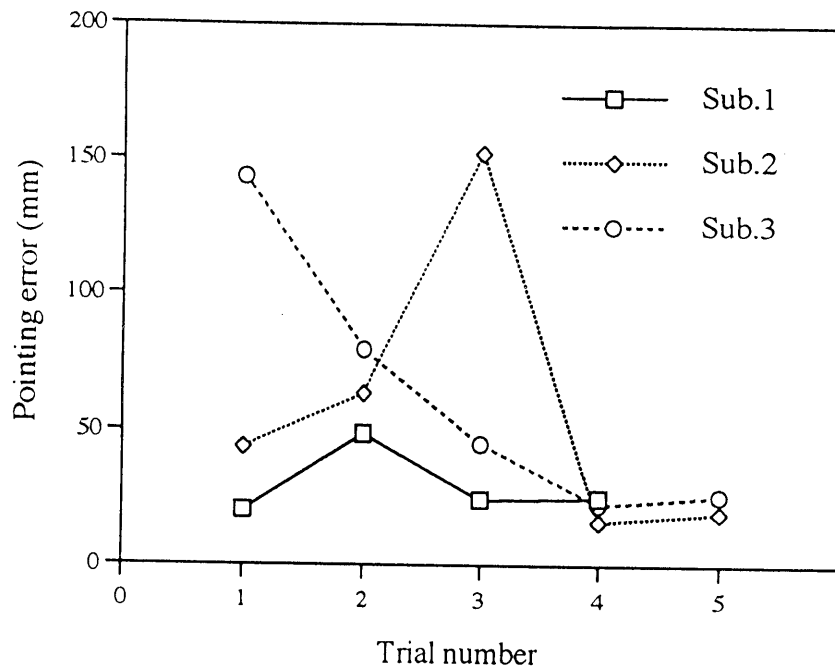


図 4.13 位置認識の誤差

8点の内2回大きな誤差が現れているためである。この2つの大きな誤差を取り除くと、平均は 48.3 mm となり、学習が順調に進んでいることが分かる。Sub. 3 は、試行回数が増えるに従って順次誤差が減少している。どの被験者も試行回数 4～5 回のところで、誤差の平均が 20 mm ～ 26 mm の範囲に収まっている。この結果は実験# 2 (目標物体、カーソルともに実物体で、3 D の環境) に近い誤差となっている。

誤差の平均値で議論を進めると、本研究における立体視環境は、従来の 2 D 環境に比べ奥行き認識においても大きな改善がなされていることを実験的に示すことが出来た。

次に、位置認識の誤差のデータを個別に考察する。表 4. 3 において、試行回数 4～5 の時最大誤差を調べると、Sub. 1 においては 70 mm、Sub. 2 においては 70 mm、Sub. 3 においては 87 mm の誤差が観察される。これらの大きな誤差がでるが、原因として次のことが考えられる。

位置認識の実験中に、仮想の3次元の空間上で、視点とカーソルと目標点がほぼ同一線上になる時には、カーソルとターゲットの図形が重なるために、2つの位置関係がつかみにくくなるものと思われる。このように一度両者の位置関係が判別出来なくなると、その後の位置認識に影響を与え、大きな誤差を生む原因となっている。

逆にカーソルとターゲットが、ある程度の距離を保っている場合、つまり視点とカーソルと目標点が同一線上からある程度離れている時には、両方の奥行き方向の差がはっきりと認識できることが分かった。したがって、この実験においては、目標点に近づく時には、まず両者の奥行き方向をある程度同じにするように接近する。その後、両者が重なる方向に接近することによって良好な認識が可能となることが分かった。

大きな誤差を生む場合の解決方法として入力装置の改善が考えられる。本実験では、テンキーによる入力を採用している。この入力方法は、目標点とカーソルがある程度接近し、微調整の段階に入ったときには良好な結果を生むと考えられる。一度目標点とカーソルの位置関係が分からなくなった場合には、カーソルを目標点から速やかに離して、大局的に両者の位置関係の認識を立て直す必要がある。この迅速な行動をとるにはテンキー入力は不利なものとなる。例えば、マウスを利用した方法を採用すれば、カーソルの移動速度をテンキーに比べて速くすることが出来るので、位置認識を立て直すための行動が取りやすく、従って、位置関係が把握し易くなり、大きな誤差は出にくくなると考えられる。

図4.14は、実験#1～#4までの結果をまとめたもので各被験者の認識誤差の平均値を図に表したものである。#1は立体視を行っていないので、非常に大きな誤差が現れている。実験#2は、立体視環境のもとでの2つの実物体（円柱）間の位置認識の誤差を表している。平均値は20mmである。実験#1に比べ、立体視環境が与えられた場合、奥行き方向の位置認識が顕著に改善されることを示している。しかし、人間が自分の目で物を直接みて位置の認識をする場合、その誤差は少なく見積もっても5mm以内にはなるもの

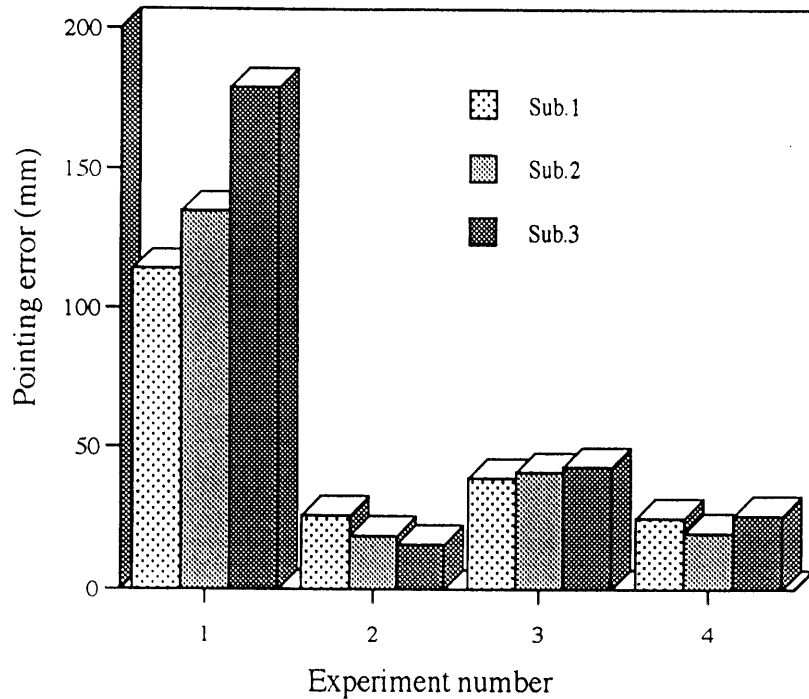


図4.14 位置認識の誤差の比較

と推定されるので、これに比べ本研究で用いた立体視環境はまだ改善の余地が残っている。

実験#3は、実物体を目標にして仮想カーソルを目標物に出来るだけ近づける実験である。この実験での誤差の平均値は41mmであった。実物体どうしの位置認識の約2倍となっている。この誤差の原因として考えられることは、次に述べる2つの事柄である。まず一つは、実空間と仮想空間の対応誤差である。この実験では、実空間と対応する仮想空間を構成する必要があり、透視変換マトリクスを測定している。グラフィック表示はこの測定された透視変換マトリクスを用いて、実空間と重ね合わせられる。よって、被験者にはカーソルが目標物の所に確実に移動していると認識されても、2つの空間が正確に対応していなければ誤差が生じることになる。次に考えられる原因は、実物体と仮想物体の表示の違いによる誤差である。実物体は、CCDカメラで撮影された映像で表現されているため、光の照射による色の濃淡や、物体の質感などが表現されているが、一方仮想カーソルは、



グラフィックによる線画であるため、映像に比べ含まれている情報量に大きな差が生じている。このため両者間の位置認識において、情報量の差が誤差として現れているものと考えられる。

最後に、実験#4について述べる。この実験は、目標物もカーソルも仮想空間に描かれたものである。したがって、実験#3における、実空間と仮想空間との対応誤差は考える必要が無い。また、両者は、グラフィックによる線画で表されているので、映像の情報量にも差は無いものと考えられる。平均誤差は24mmとなっている。これは実験#2の値に近いものとなっている。本研究で用いられた立体視環境において、実物体間の位置認識が最も良くなることは、当然の結果であるので、それに続く良好な結果が出ていることは、仮想世界での位置認識も良好に行われることを示唆している。

#### 4.5 結 言

仮想平面を使った障害物回避として、経路修正法と曲線追跡による回避の方法を提案した。両法とも、その有効性をシミュレーションまたは実験を行って確かめた。

つづいて、立体視環境における3次元空間の位置認識について、その改善の程度を実験的に確かめ良好な結果を得ることができた。

## 第5章 仮想センシング法を用いた複数ハンドロボットの衝突回避

### 5.1 緒言

この章では、複数のハンドロボットの同時操作における衝突回避のための仮想センシング法について述べる<sup>(17) (18) (24) (29) (30) (32) (33) (34)</sup>。

ハンドロボットの衝突回避においては、複数の物体の衝突判定について、人間が日常的に行っている視覚的判定法を応用した新しい方法を提案している。この方法は、数学的手法はほとんど用いられず、ディスプレイに呈示されたハンドロボットのグラフィック表示をもとに判定されるものである。衝突判定の過程が逐次人間に呈示されるので、不測の状況に陥った時は、直ちに人間が介入することが可能であり、システムの運用における保守性に富むものとなる。

両腕ロボットの衝突回避問題は、従来はロボットを構成する各リンクの3次元位置と姿勢の情報を用いて数学的に解かれてきた。ここで、提案している方法は、人間が本来行なってきた視覚的判定法を応用したものである。人間はその判定に際し、数学的計算を行なうことなく、人間の目を通して与えられる平面的映像のなかで、二つのロボットが、重なっているかどうかでその衝突状況を判定している。しかし、与えられた映像のなかで、二つのロボットが重なっていても、必ずしも衝突しているとは言えないことを経験上知っている。そこで、その判定を確実なものにするために、自らの視点をロボットの周辺で、移動させることによって平面映像上の判定を確実なものにしている。ここで提案している方法はこの一連の人間の判定行動をシミュレートすることによって衝突の判定ができることを示そうとするものである。この方法は、実ロボットと全く同じ形状をした仮想ロボットを用いて行なわれ、この仮想ロボットのある平面への投影図だけを用いて衝突の判定を行

なうことに特長がある。

## 5.2 仮想センシング法のコンセプト

本研究では、ここに「仮想センシング法」と名付ける新しい方法を提案している。この方法の特徴は、可能なかぎり数学的方法を避けようとするところにある。図5.1(a)の右図は従来からの数学的手法を用いた方法を説明している。この方法では、(1) ロボットの姿勢と(2) 衝突の判定と(3) 衝突回避の方法を数学的に取り扱っている。そして、全ての結果が得られた時点で、ロボットの様子をディスプレイに表示して人間に呈示される。

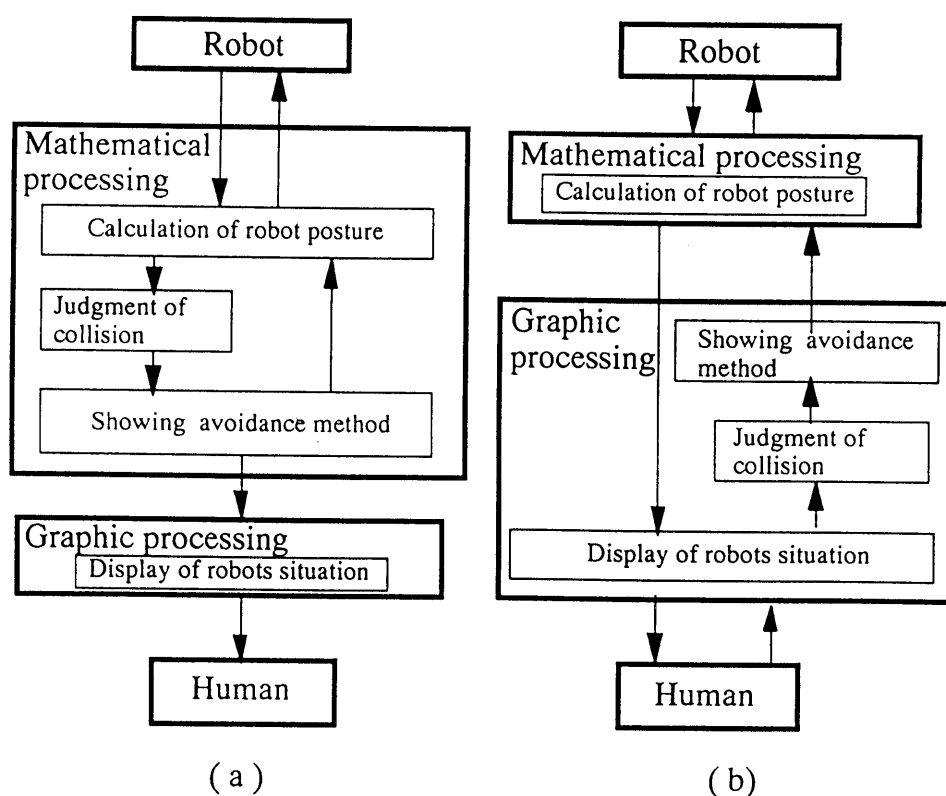


図5.1 仮想センシング法の概念

つまり数学的処理の途中経過は提示されず人間オペレータの仕事はただ観察するだけとなる。一方図 5.1 (b) は、新しく提案した仮想センシング法を説明している。この方法では、数学的に処理をしているのは、ロボットの姿勢の計算だけである。ロボットがある指定された位置へ移動する時、エンドイフェクターの先端座標を指定すると各関節の角度が計算されロボットの姿勢が決定される。この部分だけが数学的に処理されるだけで、以降の衝突の判定や衝突回避の方法は、グラフィックスを用いて処理され、その過程はいつも人間に呈示されるのである。それゆえに、人間オペレータは衝突の判定、衝突の回避法をたえず観測することができる。さらに、グラフィックスを通してロボットの制御に介入することもできるのである。つまりこのシステムの中での人間は、単なる監視者ではなく不測の状況が起こった場合は、ロボットの制御に積極的に介入することが出来るのである。このことによって、衝突回避システムの信頼性と保守性を高めることができる。

### 5.3 仮想センシング法の手順

図 5.2 は、仮想センシング法の手順を説明したものである。まず、ある時刻において、ロボットの姿勢が計算されたものと仮定する。そこで、一方のロボットを含む平面を仮定し、その平面で他方のロボットを切断する。切断面の求め方は後で説明する。切断されたときの他方のロボットの断面形状が得られると、その断面形状の相似形をした仮想近接センサがディスプレイに描かれ、接近の程度がこれらのセンサを用いて判定される。衝突判定は次のように行なう。

図 5.3 においてロボット I を表している線分が、仮想センサの線分によって切断されているかどうかで判定される。例えばロボット I の線分を白色に仮想センサの一番外側の線分を赤色に、2 番目の線分を黄色と仮定する。もし衝突をしている場合は、ロボット I

を表している白い線分が、赤色や黄色の pixel で切断されていることになる。つまり、ロボット I を表現している白色の線分に沿って赤色や黄色の pixel が存在しているかを探索することによって、衝突の位置や接近の程度が分かり衝突判定が可能となる。

以下に手順を詳述すると

- (1) ロボット I のある姿勢を表示する
- (2) ロボット II の断面形状と仮想センサを描く
- (3) ロボットを構成するボディ、アッパーアーム、フォアアーム、エンドイフェクターの順に白色の線分が赤や黄色の pixel で切断されていないかを調べる。
- (4) どの部分においても切断されていなければ、衝突は起こっていないことを表すことになる。
- (5) 切断部分が検出された場合、どの矩形において検出されたかによって、ロボット I

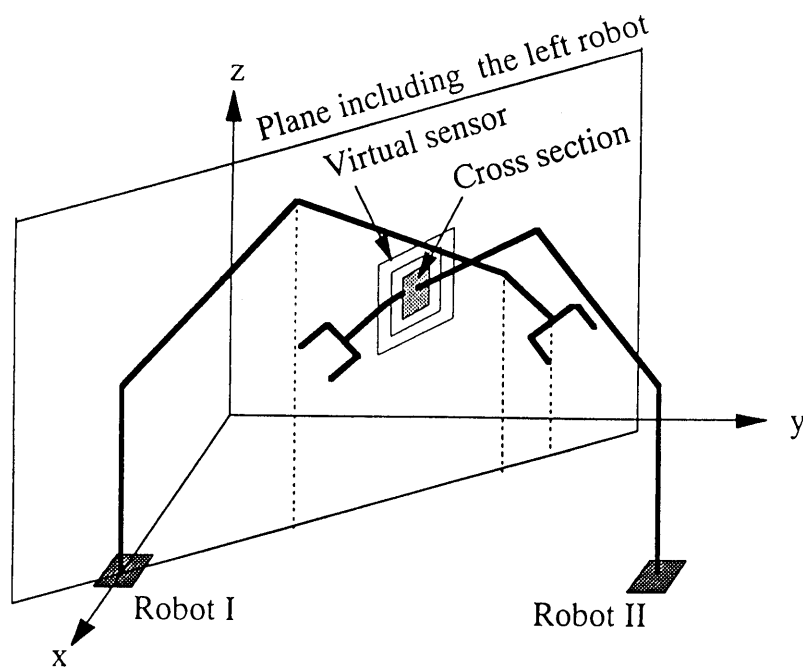


図 5.2 仮想センシング法の手順

のどの部分と衝突の可能性があるかが判定できる。さらに、検出された pixel の色によって接近の程度も判定される。

上述した方法を、衝突可能な範囲において繰り返すことにより、2台のロボットが協調作業を行なう際の衝突回避問題を解決することができる。

さらに、コンピュータのディスプレイ上において、探索された赤色または黄色の pixel の2次元座標  $x^*$ ,  $y^*$  が求められれば、次のようにして衝突部の3次元座標  $x$ ,  $y$ ,  $z$  を求めることもできる。

第3章 3.3.2節の、「仮想平面上の任意の点の2次元座標から3次元座標への変換」において、ディスプレイ上の2次元座標が3次元座標に変換できることを述べたが、この考え方を導入することによって解決できる。ロボットIを含む平面を仮想平面にみたてることにより、式(3.27), (3.28), (3.29)がそのまま使用可能となる。これらの3つの式は、 $x$ ,

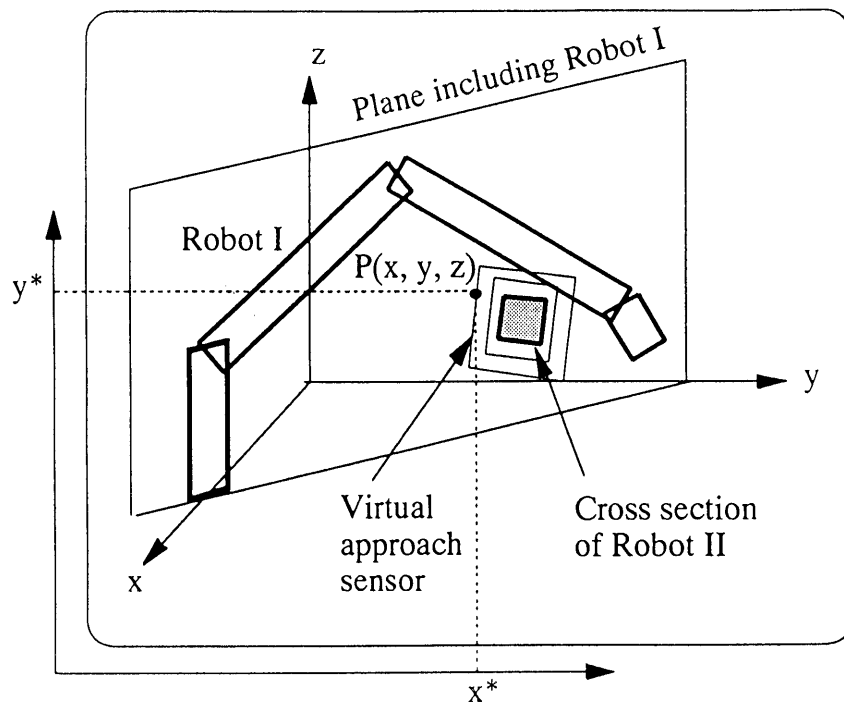


図5.3 衝突の判定

$y, z$  の連立方程式になるので、これを解くことにより3次元座標を求めることができる。

つまり、ロボットIを表す矩形を構成する線分の切断点の2次元座標  $x^*, y^*$  から作業空間における衝突場所の3次元座標  $x, y, z$  が求められる。

次に、ロボットアームを構成するリンクの断面形状を求める方法を図5.4に示す。断面の形状は、数学的方法は一切用いられず、機械系の技術者がドラフターを用いて製図板上で図を描くときのように、すべて図形的処理を用いて断面が求められる。これらの2つの図はそれぞれ、アームの正面図と側面図である。図中の線分CDは、前の図で説明した左のロボットを含む面を表している。この面が描かれると、即座に側面図上に線分CDで切り取られるアームの断面が描かれる。

この断面図は、ディスプレイに描かれた図形から、任意の場所の断面図を描くことができる。例えば、正面図として描かれた図形の線分AB（赤）と切断面を表す線分CD（白）の交点 $P_1$ は次のようにして求められる。線分CD（白）が描かれた後、線分AB（赤）を描くと線分CDは、赤色点で切断されたことになる。つづいて、線分CD（白）上で赤色

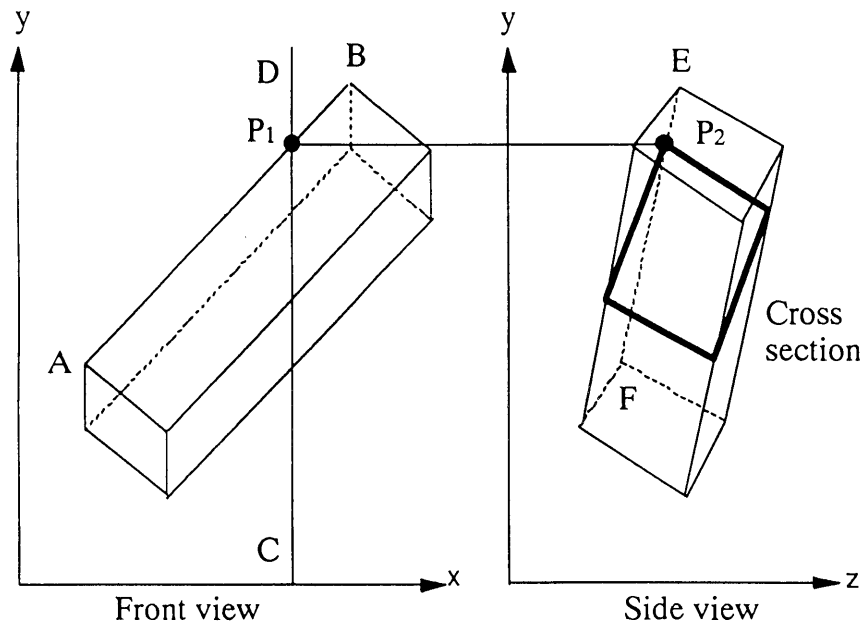


図5.4 ロボットアームの断面形状を求める手順

の点を探索することによって二つの線分の交点 $P_1$ の座標は求められる。 $P_1$ が求められると、 $P_1$ から右方向に延長した線分と側面図（この図では省略されている）内の対応線分との交点 $P_2$ を求めることができる。これらの作業を続けることによって、アームの断面図が求められる。上記の手法を用いて断面形状を求めた一例を図5.5に示す。この図は、コンピュータのディスプレイ上に表示されたものである。2つのハンドロボットの、ある姿勢での正面図と側面図を重ねて表示している。これは人間が衝突判定を行う場合に、視点を変えて対象物を観察することに対応している。左図は、断面の形状を求める前のものであり、この図からは衝突の判定をすることはできない。しかし、ロボットIを含む平面で切断されるロボットIIの断面形状が求められれば衝突の判定が可能となる。右図は断面形状を求めた後の図であり、4つの小円を結んだ4辺形が求められた断面形状を表している。この図において、側面図のロボットIとロボットIIの断面図が重なっていることから、2つのハンドロボットが衝突していることを判定することができる。

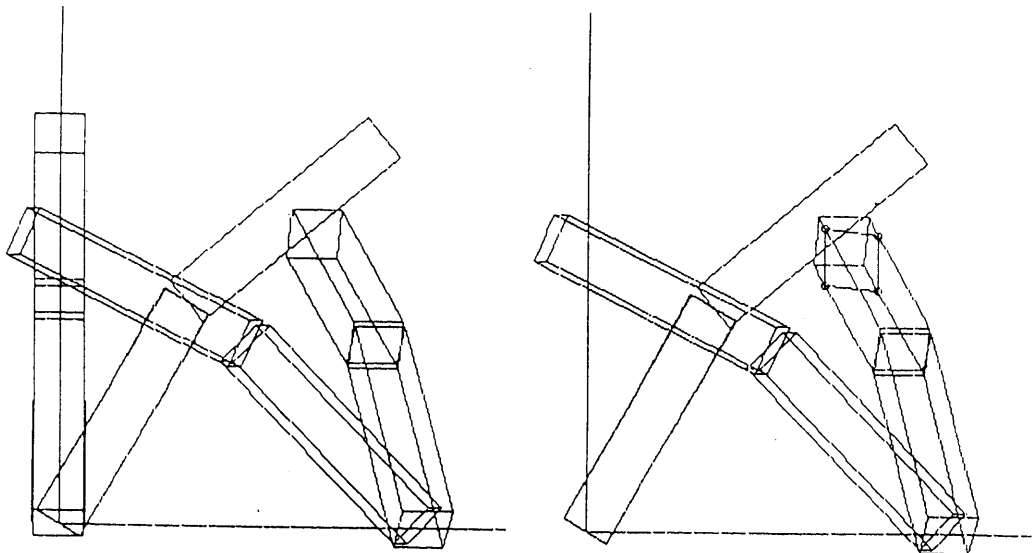


図5.5 断面形状取得と衝突判定の一例



## 5.4 衝突回避行動

ロボットIIの断面図とその周りに配置された仮想センサが、描かれると衝突の判定の他に、衝突回避の行動をとることが出来る。図5.6、図5.7は衝突回避行動を説明したものである。まず図5.6は、ロボットIのフォアアームの下部にロボットIIが位置していることを示している。フォアアーム部を表す矩形の線上を探索することによって、フォアアームの下部に、障害物が位置していること、そしてこの障害物との接近の程度は衝突寸前であることが分かる。何故ならば、矩形を構成する下部の線を黄色のpixelが切断していることが分かるからである。この場合、フォアアームを上部に移動させなければならない。衝突回避の方法には次の3つのモードが用意されている。

### (1) エンドイフェクターの角度を変える場合

エンドイフェクターの角度をマイナス方向に増加させることによって、障害物回避空間を増加させることが出来る。ただし、ここでの障害物回避空間とは、ロボットIのボディーの底部とエンドイフェクターの先端を結ぶ直線とロボットのリンクで構成される5角形の閉空間の面積である。このモードでまず回避を試みる。このモードの持つ重要な意味は、エンドイフェクターの先端の座標を変えずに、回避行動が出来ることである。エンドイフェクターの先端の座標は、ゴールへ移動するための道筋であるので、この道筋を変化させずに障害物を回避できることは重要な意味を持つのである。

上述の状況と反対に、障害物がロボットIの上部に認められるときは、エンドイフェクターの角度をプラス方向に増加させることによって、回避行動を行う。

### (2) (1)のモードで回避できない場合

障害物が下部にある場合は上方に、反対に障害物が上部にある場合、下方に、エン

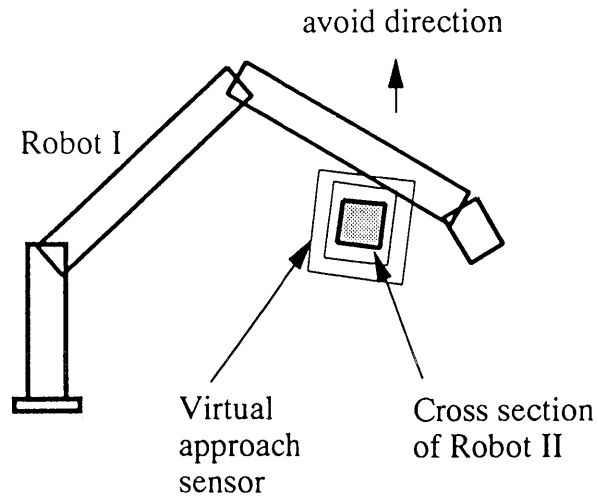
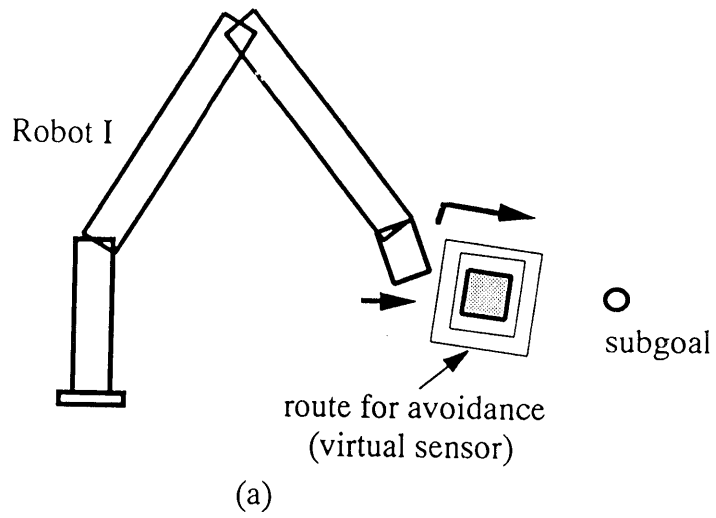
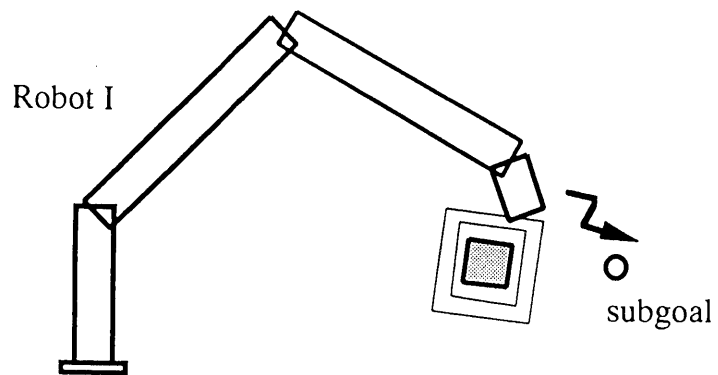


图 5.6 衝突回避行動(I)



(a)



(b)

图 5.7 衝突回避行動(II)

ドイフェクターを移動させて、回避を行う。

### (3) 仮想センサを回避ルートに利用する場合

エンドイフェクターとサブゴールの間に障害物がある場合は、仮想センサを回避ルートとして、利用することによって衝突回避を行う。この様子を、図5.7に示す。エンドイフェクターがサブゴールに向かって移動中に、エンドイフェクターの一部が仮想センサに接触したことが検出されると、仮想センサ自身が、衝突回避のルートとなって、エンドイフェクターをサブゴールに導くことが出来る。仮想センサを回避ルートとしてエンドイフェクターが追跡する方法は第4章4.3.1節に詳しく述べられている。

## 5.5 考察

この章で提案した仮想センシング法を検証するために次のような処理の実行速度を検討した。なお、検討に用いられたコンピュータはNEC社のPC-9821Xnである。その結果は、ロボットの断面形状および矩形の4つの頂点の3次元座標を求めるのに11 msecを要し、衝突判定のための矩形の全周の探索に19 msecを要した。2台のロボットの衝突によって、途中で断面矩形の図が切断されていることが判定された場合は、この時間よりもっと短い時間で処理することができる。衝突判定に必要なトータルの時間は最大で30 msecとなった。衝突判定はロボットの位置の移動にともなって、何回も必要となるが、この時間はそれに充分耐えられる時間であった。なお、検証に用いたコンピュータは、安価に手に入るパーソナルコンピュータであるので、システム全体をパーソナルコンピュータレベルの計算機で構成できることも明かとなった。

## 5.6 結 言

複数のロボットの協調作業の現場として果実の自動収穫システムを考えているが、このとき、最も重要なことはロボットどうしの衝突の回避である。ここで提案した仮想センシング法は、人間が本来行ってきた視覚的判定法を応用したものである。この方法は、ブラックボックスになりがちな数学的手法をなるべく避け、人間に理解しやすいグラフィックを多用したものとなっている。あたかも機械技術者が製作図面を描くような方式で、衝突の判定をしようとするものである。予測外の事態に陥り、緊急の修復が望まれるとき、ここで提案した仮想センシング法を用いれば、システムのどの部分が故障の原因となったかが、推定しやすいものとなっている。つまりシステムの保守性にも配慮したものとなっている。

## 第6章 結 論

本論文では、ハンドロボットの遠隔操作におけるマンマシン・インタフェースの改善を図るために、3次元計測システムと仮想平面を用いた6自由度のハンドロボットの2次元制御のシステムを構築した。なお、このシステムは、時分割液晶眼鏡を用いることによって、人間に立体視環境を与えることも出来る。つづいて、作業環境のなかに存在する障害物を回避する手法を新しく提案した。さらに、複数ハンドロボットを同時に用いる場合の衝突回避のための新しい提案を行った。本研究で得られた成果を各章ごとに要約すると以下のようなになる

第2章では、まず、3次元計測の原理と手順を説明し、以下の実証実験に用いられ有用であることが確かめられた。透視変換マトリクスの測定については、以下の結果が得られた。既知点16点を用いて透視変換マトリクスの各要素を求めた場合、座標軸の設定には、その影響はほとんど無いことが確かめられた。つまり、この場合は、どの方向から撮影しても精度の高い透視変換マトリクスが求められることが分かった。既知点8点を用いた場合、最小誤差と最大誤差の比較において、約3倍の差が生じていることが分かった。カメラの撮影方向に対してx軸回りの角度  $\theta=0$  の時 (y軸回りの角度  $\phi=0$  または  $\phi=90$ )、または  $\theta=90$  の時 ( $\phi=0$  または  $\phi=90$ ) には、16点法と同レベルの低い誤差となっている。しかし、この状態にカメラを精度良く設定するには非常に時間がかかるので、16点法を用いた方が、既知点の入力に少々時間がかかるものの、カメラの姿勢に依存しない精度の良いマトリクスが求められるので、高精度の測定が必要なときは16点法を採用する方が良いことが分かった。

つづいて、エンドイフェクターに取り付けたカメラによる実験結果では、測定誤差は最大で 1.9 mm であり、固定カメラとエンドイフェクター上のカメラの両方を用いたシステ

ムの実験結果では測定誤差の最大は y 方向の 24.4 mm であった。今回の実験によって、補正を行えば最大で 11.6 mm の測定誤差の範囲内で 3 次元測定が可能であることを示すことが出来た。

4 台のカメラを用いた、2 段階の計測法により、約 10 mm 前後の誤差でハンドロボットが対象物を取りにいけることを、実験的に確認することが出来た。エンドイフェクターに取り付けたカメラの固定法をもう少し堅固なものにすることが出来れば、さらに良い精度でハンドロボットを制御できるものと考えられる。これらの結果より、作業環境から遠く隔たった所にいるオペレーターが、作業環境の中にある任意の対象物の 3 次元位置を良好に計測出来ることを実験的に示すことが出来た。

このことより、ハンドロボットで遠隔操作をする場合、呈示された映像を見ながらハンドロボットのエンドイフェクターを逐次近づけるのではなく、得られた 3 次元情報をもとに直接対象物を掴む動作が可能となり、オペレーターの負担を軽減することができるようになった。

第 3 章では、人間に立体視環境を呈示できる遠隔操作システムの構築、仮想平面を用いたロボットの制御について述べた。

仮想平面を用いたロボット制御においては、システムへの入力状況を仮想空間の中に操作空間として表示した。操作空間は、仮想平面とマウスの組み合わせによって構成されている。マウスは 2 次元の入力装置であるが、空間の中で任意の方向に回転出来る仮想平面を設置することによって、6 自由度の制御が可能となった。

さらに、ロボットが作業をする実空間と、操作空間としての仮想空間をディスプレイ上にスーパーインポーズして表示することによって、インタフェースの改善を図った。このことにより、人間の感覚にマッチした形態で入力と出力の両者の情報を提示することが可能となった。例えば、作業空間の中のハンドロボットを、ある位置へ移動させる場合、そ

の位置にマウスを移動させて指示するだけで、ハンドロボットを所望の位置に移動することが出来る。また、そのとき立体視環境は、人間の位置認識の能力を高めることに貢献している。

第4章では、まず、経路修正法による障害物回避の方法を提案し、シミュレーションでその有用性を示した。これは、エンドイフェクターが物体を把持した状態で、障害物を回避しながらゴールへ到達させる経路を見つける方法である。

つぎに、仮想平面上に描かれた曲線を用いた障害物回避においては、平面上に描かれた任意の曲線を追跡することによって、ハンドロボットのエンドイフェクターの先端部の障害物回避が可能であることを実験的に確かめることが出来た。このことより、障害物回避のための経路をグラフィックで描いて仮想空間の中に設置することによって、実ロボットの障害物回避が可能となることを示すことが出来た。

位置認識実験において、従来の2D環境に比べ奥行き認識においても大きな改善がなされていることを実験的に示すことが出来た。

第5章では、仮想センシング法を用いた複数ハンドロボットの衝突回避について述べている。本研究の応用として、自動果実収穫システムへの導入を考えているが、このとき、最も重要なことはロボットどうしの衝突の回避である。ここで提案した仮想センシング法は、人間が本来行ってきた視覚的判定法を応用したものである。この方法は、ブラックボックスになりがちな数学的手法をなるべく避け、人間に理解しやすいグラフィックを多用したものとなっている。あたかも機械技術者が製作図面を描くような方式で、衝突の判定をしようとするものである。予測外の事態に陥り、緊急の修復が望まれるとき、ここで提案した仮想センシング法を用いれば、システムの故障原因や故障個所の推定が容易なものとなる。つまりシステムの保守性にも配慮したものとなっている。

## 参考文献

- (1) T.B. Sheridan "Telerobotics", 10th IFAC World Congress on Automatic Control, 1987.
- (2) 佐藤知正 "テレロボティクス研究の動向", 日本ロボット学会誌 Vol.11 No.6, pp.777-781, 1993
- (3) Susumu Tachi and Mititaka Hirose, "バーチャル・テック・ラボ(超現実への接近)", 工業調査会.
- (4) Susumu Tachi, "テレロボティクスの世界", 日本ロボット学会誌 Vol.11 No.6, pp.770-772, 1993.
- (5) Sheridan, T. B., "Human Supervisory Control of Robot System", IEEE Conf. Robotics and Automation, pp808-812, 1986.
- (6) Sheridan, T.B., "Telerobotics, Automation and Human Supervisory Control", The MIT Press, 1992.
- (7) 佐藤方彦 "マンマシン・インターフェイス", 朝倉書店.
- (8) Das, H. et al. , "Computer Aided Path Planning for Redundant Teleoperators", NATO Advanced Research Workshop on Robots with Redundancy, June 27, 1988.
- (9) Sheridan, T. B. and Inoue, S., "Use of Graphic in Decision Aids for Telerobotic Control", Proceeding of the Work Shop on Space Telerobotics, NASA, 1989.
- (10) 橋本孝之, Sheridan, T.B., Noyes, M.V., "時間遅れを有するテレオペレーションにおける予告情報の効果", 人間工学, Vol.22, No.2, 1986.
- (11) Rosenberg, L. B., "Virtual Fixtures : Perceptual Tools for Telerobotic Manipulation", Proceedings of VRAIS '93, pp.76-82, 1993.
- (12) Milgram, P. and Drascic, D., "Enhancement of 3-d Video Displays by Means of Superimposed Stereo-Graphics", Proceedings of Human Factors Society 35th Annual Meeting, pp.1457-1461, 1991.
- (13) Drascic, D. and Milgram, P., "Positioning Accuracy of Virtual Stereographic Pointer in a Real Stereoscopic Video World", SPIE Vol.1457 Stereoscopic Displays and Applications II, pp.302-313, 1991.



- (14) Milgram, P. et al., "Applications of Augmented Reality for Human-Robot Communication", Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robot and Systems, pp.1464-1472, 1993.
- (15) Inoue, S. et al. "Two Dimensional Control for 6 d.o.f. Hand Robot Teleoperator", 2nd IEEE International Workshop on Robot and Human Communication, pp.171-176, 1993.
- (16) Inoue, S., "Path Modification for Three Dimensional Obstacle Avoidance", Research Reports of Fukui National College of Technology, 23; pp.69-74, 1989.
- (17) Inoue, S. et al. "Cooperated Operation of Plural Hand-Robots", Proceedings of 14th IMACS World Congress on Computational and Applied Mathematics, Vol. 3, pp.1326-1328, 1994.
- (18) Inoue, S. et al. "Cooperated Operation of Plural Hand-Robots for Automatic Harvest System", International IMACS Journal "Mathematics and Computers in Simulation" (July 1996 掲載予定) .
- (19) Rogers, D.F. and Adams, J.A., "Mathematical Elements for Computer Graphics", McGraw-Hill, 1976.
- (20) 井上清一, 橋本孝之, "任意位置にある1台のカメラによる3次元情報の計測", 福井高専研究紀要, 第20号, pp.111-116, 1986.
- (21) 井上清一ほか, "遠隔操作におけるマニピュレータの運動情報の呈示に関する研究", 日本機械学会北信越学生会卒業研究発表会講演前刷集, pp.403-404, 1988.
- (22) 佐藤知正, "視覚・力覚を手元に集中させた遠隔微細操縦システム", 日本機械学会第70期通常総会講演会講演論文集(III), pp.445-447, 1993.
- (23) 竹元宜弘ほか, "Cによる数値計算", 朝倉書店, 1987.
- (24) 今井隆, "果実の自動収穫システムの開発に関する研究", 岐阜大学大学院工学研究科修士論文, 1994.
- (25) 小林孝浩, "テレロボティクスの構築とその応用に関する研究", 岐阜大学大学院工学研究科修士論文, 1994.
- (26) 小鹿丈夫, 井上清一ほか, "バーチャルリアリティに関する研究(I) -テレロボティクス-", 岐阜大学工学部研究報告, 第43号, pp.53-59, 1993.
- (27) 小鹿丈夫, 井上清一ほか, "バーチャルリアリティに関する研究(II) -テレロボティクス-", 岐阜大学地域共同研究センター研究成果報告, No. 3,

- pp. 83-90, 1993.
- (28) 小鹿丈夫, 井上清一ほか, "バーチャルリアリティに関する研究(III)  
-テレロボティクス-", 岐阜大学工学部研究報告, 第44号, pp. 75-84, 1994.
  - (29) 小鹿丈夫, 井上清一ほか, "バーチャルリアリティに関する研究(IV)  
-テレロボティクスとバーチャル鶴飼システム-", 岐阜大学地域共同研究センター  
研究成果報告, No. 4, pp. 65-74, 1994.
  - (30) 箕浦弘人, 小鹿丈夫, 井上清一ほか, "バーチャルリアリティに関する研究(V)  
-テレロボティクスにおける協調動作-", 岐阜大学工学部研究報告, 第45号,  
pp. 43-50, 1994.
  - (31) 小鹿丈夫, 井上清一ほか, "バーチャルリアリティに関する研究(VI)  
-テレロボティクスとバーチャル鶴飼システム-", 岐阜大学地域共同研究センター  
研究成果報告, No. 5, pp. 73-80, 1995.
  - (32) 小林孝浩, 小鹿丈夫, 井上清一ほか, "果実収穫支援システムの開発(1)", 岐阜大学  
工学部研究報告, 第45号, pp. 37-42, 1995.
  - (33) 小鹿丈夫, 井上清一ほか, "バーチャル鶴飼システム(1)", 岐阜大学工学部研究報告,  
第45号, pp. 59-64, 1995.
  - (34) Kobayashi, T., Inoue, S. et al., "Development of a Fruit Harvesting System",  
Proceedings of International Conference on Virtual System and Multimedia  
'95, pp. 295-301, 1995.

## 謝 辞

本論文を結ぶにあたり、終始御懇篤なるご指導とご鞭撻を賜りました、岐阜大学工学部電子情報工学科、小鹿丈夫教授に深甚なる謝意を捧げます。

福井工業高等専門学校田中茂利校長には、本研究の機会を与えて下さったばかりでなく、終始ご激励を賜りました。ここに心から感謝致します。

大阪教育大学橋本孝之教授には、研究遂行全般にわたり、ご指導とご鞭撻を賜りました。ここに、衷心より感謝の意を表します。

大学院工学研究科小鹿研究室の学生諸氏には多大なご協力を頂きました。心より感謝致します。特に、第3章及び第5章の研究成果は、小林孝浩氏、今井隆氏（現三陽電機製作所(株)）との共同研究より得られたものであります。青木克央氏と箕浦弘人氏には、本研究におけるシステムの設計と構築に多大のご援助を頂きました。ここに、深く感謝致します。また、実験装置の製作や実験において、福井工業高等専門学校機械工学科の卒業研究生諸氏のご協力を頂きました。ここに、深く感謝致します。

平成8年1月