

学位請求論文

190

センサ情報に基づく自律型移動ロボットの  
行動制御に関する研究

平成2年12月21日提出

指導教官  
原島 文雄 教授

東京大学大学院 工学系研究科  
電気工学専攻 87101  
久保田 孝

---

学位請求論文

---

センサ情報に基づく自律型移動ロボットの  
行動制御に関する研究

東京大学大学院 工学系研究科

電気工学専攻 87101

久保田 孝

# Study on Behavior Control for Autonomous Mobile Robot

## Based on Sensor Information

*By Takashi KUBOTA*

Sensor based robot control system can overcome many of the difficulties of uncertain models and unknown environments which limit the domain of application of current robots used without external sensory feedback. Both industrial robotic arms and mobile robots require sensing capability to adapt themselves to new tasks without explicit intervention or reprogramming. These relationships between sensing and control have been long recognized. With the help of sensors, a robot can have an "*adaptive behavior*" : the robot is able to deal properly or flexibly with changes in its environment and to execute intelligent tasks.

Recently researches on intelligent mobile robots have attracted much interest. To realize intelligent robots, it is necessary to provide a robot with many facilities, such as the facility of recognizing environment and the ability of planning tasks etc. Planning a collision-free path is one of the fundamental requirements for a mobile robot to execute its tasks. Especially path searching in a 2D world with polygonal obstacles has received considerable attention as a part of the general problem of mobile robot motion planning, and various algorithms have been proposed. However there are few indoor guidance systems that can create a path plan in an acceptable period time and execute it in practically complicated environment by detecting

and avoiding unknown obstacles. Most of path planning schemes which concern obstacle avoidance concentrate on the problem of avoiding fixed and stationary obstacles in a workspace. Realistically, however, obstacles are not always stationary. Considering the situation where several robots operate together, it is necessary that a robot can deal with moving objects.

In recent years, there has been increasing interest in upgrading robot intelligence by using multiple nonvisual sensors as well as visual sensors. Advanced robotics applications demand a steady stream of rich and reliable information to be gleaned rapidly from sensors, because they are required to operate in an uncertain environment. Recently more and more efforts have been put in the fusion or integration of multiple sensors into robot systems. Therefore this paper also focuses on the multiple sensor system, especially sensor integration or fusion techniques. Many researchers have proposed various kinds of multi-sensor systems. Though many good ideas have been explored, several key issues remain to be resolved to make possible a more general multi-sensor system. There are few sensor fusion systems to integrate multiple sensor information in real time at the control level. Here sensor fusion need to be considered in view of control of mobile robot.

This thesis mainly consists of three phases: path planning, control, and behavior decision.

At the first phase, path planning problem for a mobile robot is discussed. The proposed mobile robot system has a hierarchical path planning system which searches path efficiently in an uncertain environment. The planning system consists of a global planner and a local planner. The global planner gives a global path in terms of a sequence of visual sub-goals. Then the local planner

generates a local path between the sub-goals with the help of visual sensor. The main focus of this phase is on the local path planning which provides real-time guidance to the system. Visual sensor can provide useful information about the environment. So an algorithm is proposed to generate avoiding points by using visual information to bypass unknown obstacles in the local path planning. The local path planning in simple environment is simulated by using three dimensional graphics. A simple experiment is also done for the case that there are two obstacles. The effectiveness and validity of the proposed method is verified by these simulations and experimental results.

At the second phase, navigation problem for a mobile robot is discussed. Among various sensors, vision is expected to play a very important role in control of intelligent mobile robot. Firstly a method is presented to guide the robot in real time by using visual sensor. The robot is controlled to follow visual sub-goal by visual feedback. The mobile robot can detect unknown obstacles and make avoiding paths automatically. Obstacle avoidance based on visual information is then simulated. The result of the simulation shows that navigation using only visual information has some problems. To solve these problems, visual sensor and range sensors such as ultra-sonic sensors, are cooperated together. In this strategy, visual information is used mainly as global information and obtain rough information about obstacles. When detail depth information is needed, range sensors are used. It is important to integrate the information of multiple kinds of sensors. A control method, which utilizes the information fusion between vision and range sensors, is proposed.

At the third phase, decision making for a mobile robot to avoid moving obstacles is discussed. A real-time collision

avoidance method based on sensor information and fuzzy reasoning is proposed. The global states of colliding with obstacles are recognized and decision of action is made in real time. Among various sensors, vision is expected to play a very important role in control of intelligent mobile robot. In this strategy, visual information is used mainly to obtain rough information about obstacles. Virtual feature point is introduced to estimate the motion of an obstacle. The degree of collision danger is recognized by fuzzy reasoning and then action pattern is determined based on production rules. This strategy can realize a heuristic avoidance among moving obstacles.

This thesis is structured as follows. In Section 2, general concept of sensor based control system is explained and discussed. Section 3 presents a path planning method for a mobile robot to search path in an uncertain environment by using visual information. A navigation method for a mobile robot based on cooperation of vision and range sensors is presented in Section 4. In Section 5, a real-time collision avoidance strategy for a mobile robot. based on fuzzy reasoning and production rule is described. Section 6 is for discussion, conclusion, and future works.

**Key Words :** Autonomous mobile robot, Hierarchical path planning, Avoiding point, Single vision, Image processing, Visual navigation, Sensor fusion, Fuzzy reasoning. Virtual feature point, Production rule.

## 論文の要旨

論文題目      センサ情報に基づく自律型移動ロボットの  
行動制御に関する研究

氏 名      久保田 孝

産業用ロボットの利用範囲が広がるにつれて、環境の変化に順応できるような知能をもったロボットが必要となっている。ロボットが知的な動作をするためには、広範囲な環境に対して適応する能力をもつことが必要である。したがって、ロボットはまわりの状況を知るといった環境認識能力をもつことが重要であり、また自分自身が移動できるという能力をもつことも必要不可欠である。ロボットは移動能力をもつことにより、作業空間が拡大し、自ら作業環境内を自由に動いて作業を行うことができる。

移動ロボットが自律走行を実現するためには、人間が与えた知識やセンサによって獲得した情報を総合的に判断し、行動を決定しなければならない。そのためには複数の情報を合理的に処理し、目的に応じて行動を律する戦略が必要となる。その際、多種類のセンサを利用して量的ならびに質的に高いレベルの情報を引き出すことが必要であり、センサ情報の処理系が重要な課題となっている。またロボットをコントロールする場合に問題となることは、センシングによって得られた情報をいかに行動計画、制御にフィードバックするかである。従来の方法では、エンコーダ等の内界センサを利用するものが多かったが、ロボットの作業環境の拡大とともに視覚センサ、触覚センサなどの外界センサが使われるようになってきた。特にロボットシステムにおいて視覚センサの果たす役割は非常に大きいものと期待されている。しかしながら視覚情報を画像解析した上で、ロボットを制御する状態量に変換する方式が主に用いられており、

演算処理に時間がかかるなどの問題がある。

そこで、本論文は実時間処理を念頭に、移動ロボットシステムにおいてセンサから得られる情報と運動系の制御情報の整理・統合を行い、センサ情報処理系と行動制御系との有機的な結合を実現することによって、センサベースロボットコントロールシステムを構成することを大目的としている。

本論文では、以下の3つのフェーズに注目し、本研究の目的であるセンサベース移動ロボットシステムの実現のための課題を設定している。

#### [Phase\_1] 移動ロボットの行動計画

移動ロボットの経路探索問題を取り扱い、地図情報とセンサ情報を有効利用する階層型パスプランニング法を確立する。

#### [Phase\_2] 移動ロボットの制御

移動ロボットのナビゲーション問題を取り扱い、視覚センサと超音波センサとの協調を行い、センサ情報融合手法を確立する。

#### [Phase\_3] 移動ロボットの行動決定

移動ロボットの動的環境下における衝突回避問題を取り扱い、ファジィ推論によって柔軟な回避行動を実現する。

本研究において、ロボットの作業環境として屋内環境を想定し、主に視覚情報に基づいた移動ロボットの行動計画及び制御手法を構築している。屋内では、ほぼ水平な床と垂直な壁に囲まれ、その中に置かれているものも机や本棚といった人工物が多い。また自然環境に比べてイメージングの条件が良いという特徴を持っている。本論文では、移動ロボットシステムを対象とし、前述の3つのフェーズに応じてセンサ情報処理系と行動制御系との有機的な結合を試みた。

まず、Phase\_1において移動ロボットの経路探索問題を検討している。移動ロボットにおいて、スタート地点から障害物を回避しながら目標地点に到達する経路を自動的に計画することは、ロボットがそなえるべき重要な機能である。経路探索問題に関しては、多くの研究がなされており、さまざまなアルゴリズムが問題の状況に応じて提案されている。本論文では、環境に関する知識が不完全な場合の経路探索法として、視覚センサを用いた効率的な階層的プランニングシステムを提案している。このシステムは、地図画像情報からサブゴールマップを生成するグローバルプランニングと実際に単眼視による環境観測をおこなない、移動するローカルプランニングからなっている。サブゴールマップの生成法、単眼視による環境観測方法について検討を行い、センサを用いたローカルプランニングアルゴリズム、回避ポイントの自動生成法、複数障害物識別のための画



像探索法、ナビゲーション法を考案している。ローカルパスプランニング手法は、視覚センサを用いて未知の障害物を回避するためのポイントを逐次選択するアルゴリズムである。3次元グラフィックスを用いたシミュレーション、及び実験用移動ロボットを用いた複数障害物回避実験を行い、ローカルパスプランニング手法の有効性を確認している。本手法の特徴は、画像情報の大域性を利用し、リアルタイム処理が可能な程度にシンプルな点であり、実用化が可能である。

次に、Phase\_2において移動ロボットのビジュアルナビゲーションを検討している。環境が複雑になるにつれて単眼視による環境観測にはあいまいさが増大する。そこで、このあいまいさを減らすために異種センサを併用し、センサの協調を行っている。本論文では、ロボットが必要な情報を信頼性高く獲得するために、移動ロボットのコントロールという観点からセンサ情報の融合を位置づけ、複合的処理、統合的処理、融合的処理の3つの処理形態を提案している。実際に視覚センサ情報とレンジセンサ情報のベクトル合成を用いた融合方法を提案し、シミュレーションによってその有効性を示している。さらに、実験用移動ロボットを用いて、視覚センサと超音波センサの協調による実験を行い、本手法の実験的検証を行っている。センサフュージョンは、まだ新しい研究分野であり、考え方自体明確に整理されておらず、また現状では多種類のセンサを組み合わせることはそれほど簡単ではない。本手法では、移動ロボットの制御情報を異種センサである視覚センサとレンジセンサから生成しており、一種の融合とみなすことができ、この分野の研究を一步前進させたものと考えられる。

さらにPhase\_3において移動ロボットの作業環境内に複数の移動障害物が存在する動的環境を取り扱っている。移動障害物を回避するために、まず視覚センサから得られた画像データを処理することにより、移動物体のおおよその運動状態を推定している。本論文では、リアルタイム処理を念頭に、障害物回避のため仮想注目点の考えを導入している。これらの情報を基にファジィ推論によって衝突危険度を評価する。さらにプロダクションルールにより回避行動が自動生成される。本手法の有効性はシミュレーションによって示している。本手法を用いることにより人間に近い状況判断能力をロボットに与え、柔軟な回避行動を実現することができ、またルールの変更による回避性能の変化も実現できる。本手法の長所の1つは、メンバシップ関数およびルールを用いて表現することにより回避戦略が容易に記述・変更ができる点にあり、アルゴリズムのメンテナンス性が向上すると思われる。

移動ロボット以外の対象として、視覚系と制御系との密な結合を実現するという観点から、マニピュレータの位置・姿勢制御問題も検討している。目的は、画像特徴量として、対象物の特徴点4点に着目し、視覚情報を直接利用することによって、マニピュレータの先端を目標位置及び目標姿勢に移動させること

である。ここでは、幾何学的な座標変換を行わずに、並列性と柔軟性に富んだニューラルネットワークを用いることによって、画像データとマニピュレータの関節角情報との有機的な結合を行っている。シミュレーション及び実験により有効性を確認している。本手法は、複雑な幾何学的計算がニューラルネットワークの簡単なマッピングにより実現できるため、実時間処理に向いている。さらに、マニピュレータの持つコンプライアンスと組み合わせて知的な作業を行わせることが期待できる。

自律型移動ロボットに関する研究はまだ立ち上がったばかりであり、解決すべき課題が山積している。本研究は其中で重要な課題のいくつかに対しアプローチを行っている。現実世界を対象とするロボットは、センサデータの取り扱いが大きな問題であり、ロボットシステムにおけるセンサ処理系が重要な課題となっている。本研究でもこのセンサ情報の取り扱いをロボットシステム全系の中で位置づけ、そのデータ構造を上位レベルの処理に適したものにすることを念頭においている。また、センサ情報融合の考え方は、知能ロボットに限らず、自律的な機能を必要とする自動化技術において、システムの感覚系の構造を左右する重要な問題であり、この意味でもこの種の研究の意義は大きいと思われる。

# 学位請求論文

## センサ情報に基づく自律型移動ロボットの 行動制御に関する研究

平成2年12月21日提出

指導教官

原島 文雄 教授

東京大学大学院 工学系研究科

電気工学専攻 87101

久保田 孝

# 目次

1 序論	1
1.1 本研究の背景	2
1.2 本研究の目的	5
1.3 本研究の概要	7
1.4 本論文の構成	8
2 ロボットとセンサ情報	11
2.1 はじめに	12
2.2 ロボットシステム	13
2.2.1 産業用ロボットの問題点	13
2.2.2 ロボットの知能化とは	14
2.2.3 知能ロボットの基本機能	14
2.2.4 ロボットのセンサ制御系	16
2.2.5 ロボットシステムにおける情報処理	16
2.3 移動ロボットシステム	19
2.3.1 自律型移動ロボット	19
2.3.2 従来の移動ロボットシステムの問題点	20
2.3.3 自律型移動ロボット実現のための課題	21

2.3.4	移動ロボットにおける情報処理モデル . . . . .	22
2.4	移動ロボットにおける経路探索問題 . . . . .	23
2.4.1	環境に関する知識がない場合の経路探索 . . . . .	23
2.4.2	環境に関する知識が完全な場合の経路探索 . . . . .	23
2.4.3	環境に関する知識が不完全な場合の経路探索 . . . . .	25
2.5	ビジョンベースロボットシステム . . . . .	26
2.5.1	ロボットシステムにおける視覚情報の役割 . . . . .	26
2.5.2	視覚を用いた移動ロボットシステム . . . . .	27
2.5.3	マニピュレータのビジュアル制御 . . . . .	28
2.6	センサ融合システム . . . . .	29
2.6.1	センサ情報融合とは . . . . .	29
2.6.2	センサフュージョン . . . . .	29
2.6.3	センサフュージョン研究の現状と課題 . . . . .	30
2.6.4	能動型センサ融合システム . . . . .	31
<b>3</b>	<b>視覚情報を用いた移動ロボットの経路探索</b> . . . . .	<b>40</b>
3.1	はじめに . . . . .	41
3.2	移動ロボットシステム . . . . .	42
3.2.1	移動ロボットコントロールシステム . . . . .	42
3.2.2	階層型移動ロボットシステムの動作 . . . . .	42
3.3	階層型経路探索法 . . . . .	45
3.3.1	階層的プランニング . . . . .	45
3.3.2	グローバルプランナー . . . . .	45
3.3.3	ローカルプランナー . . . . .	46

3.4	グローバルパスプランニング	48
3.4.1	地図画像の利用	48
3.4.2	サブゴールマップの生成	49
3.4.3	動的計画法によるグローバルパスの一決定法	50
3.5	障害物回避動作	53
3.5.1	単眼視による環境観測	53
3.5.2	回避ポイントの決定方法	54
3.5.3	複数の障害物に対する画像処理アルゴリズム	56
3.6	ローカルパスプランニング	59
3.6.1	アルゴリズム	59
3.6.2	仮想格子点平面における適用	61
3.7	ナビゲーション	68
3.8	3次元グラフィックスによるシミュレーション	70
3.9	実験による検証	77
3.9.1	実験用移動ロボットシステム	77
3.9.2	未知障害物回避実験	78
3.10	まとめ	82
4	センサの協調による移動ロボットのナビゲーション	86
4.1	はじめに	87
4.2	移動ロボットのビジュアルナビゲーション	88
4.2.1	問題設定	88
4.3	目標追従動作	89
4.3.1	視覚情報の利用	89

4.3.2	ビジュアルナビゲーション	89
4.3.3	シミュレーションによる考察	93
4.4	障害物回避	96
4.4.1	回避経路の生成	96
4.4.2	シミュレーション	96
4.5	センサ情報処理	98
4.5.1	センサ情報のデータ構造	98
4.5.2	センサ情報の統合・融合の利点	98
4.5.3	センサ情報処理手順	99
4.5.4	センサデータのエラー検出・補償法	100
4.5.5	センサ情報の統合・融合	102
4.6	視覚センサとレンジセンサとの協調	106
4.6.1	移動ロボットにおけるセンサフュージョン	106
4.6.2	カルマンフィルタによるセンサ統合	111
4.6.3	センサベクトル合成法によるセンサ融合	113
4.6.4	シミュレーション結果	114
4.7	環境に応じた行動パターンの生成の試み	118
4.7.1	知識の表現	118
4.7.2	学習	119
4.7.3	シミュレーション結果	121
4.8	移動障害物に関する考察	124
4.9	実験による検討	125
4.9.1	実験用移動ロボットシステム	125
4.9.2	目標追従実験	126

4.9.3	センサ協調実験 . . . . .	126
4.10	まとめ . . . . .	131
<b>5</b>	<b>動的環境における移動ロボットの行動制御</b>	<b>135</b>
5.1	はじめに . . . . .	136
5.2	問題の設定 . . . . .	137
5.3	視覚センサによる障害物の検出 . . . . .	139
5.3.1	仮想注目点の設定 . . . . .	139
5.3.2	障害物の運動情報の推定 . . . . .	139
5.4	衝突危険度による状況判断 . . . . .	145
5.4.1	静的危険度 . . . . .	145
5.4.2	動的危険度 . . . . .	146
5.4.3	衝突危険度の評価 . . . . .	146
5.4.4	ファジィ推論 . . . . .	147
5.5	移動経路の生成 . . . . .	154
5.5.1	回避行動の決定 . . . . .	154
5.5.2	ナビゲーション . . . . .	155
5.6	シミュレーション . . . . .	161
5.7	画像雑音に対する考察 . . . . .	168
5.7.1	分解能に関する検討 . . . . .	168
5.7.2	画像ノイズに対する検討 . . . . .	169
5.8	他の手法との比較検討 . . . . .	172
5.9	まとめ . . . . .	173
<b>6</b>	<b>結論</b>	<b>177</b>



A	マニピュレータのビジュアル制御	181
A.1	はじめに	182
A.2	問題設定	183
A.2.1	ピック&プレイス	183
A.2.2	座標変換	186
A.2.3	画像情報からの位置・姿勢の決定	187
A.2.4	特徴点4点に基づくカメラ系の位置・姿勢関係	189
A.3	位置・姿勢制御法	193
A.3.1	ビジュアルコントロール	193
A.3.2	ニューラルネットワーク	194
A.3.3	学習・実行プロセス	195
A.3.4	学習の高速化	196
A.3.5	シミュレーションによる考察	200
A.3.6	特徴点4点の識別法	211
A.4	ニューラルネットワークシステム	214
A.4.1	学習プロセス	214
A.4.2	実行プロセス	215
A.5	シミュレーション	219
A.5.1	学習プロセス	219
A.5.2	実行プロセス	219
A.5.3	未学習パターンへの適用	220
A.5.4	ニューラルネットワークに関する考察	220
A.6	実験による検討	229
A.6.1	実験システム	229

A.6.2	ニューラルネットの学習	229
A.6.3	実験結果	230
A.7	ビジュアル制御法に対する考察	235
A.7.1	最小2乗推定法との比較検討	235
A.7.2	ニューラルネットの補間能力	239
A.7.3	ビジュアルサーボについて	241
A.7.4	冗長マニピュレータについて	244
A.8	まとめ	247
	<b>謝辞</b>	<b>251</b>
	<b>発表論文リスト</b>	<b>253</b>

# Figures and Tables

## Chapter 1. 序 論

Figure 1.1 History of robotics

Figure 1.2 Contents

## Chapter 2. ロボットとセンサ情報

Figure 2.1 Sensor control level

Figure 2.2 Hierarchical structure

Figure 2.3 Multiple sensor system

Figure 2.4 Sensor complex

Figure 2.5 Sensor integration

Figure 2.6 Sensor fusion

Table 2.1 Sensor information processing

## Chapter 3. 視覚情報を用いた移動ロボットの経路探索

Figure 3.1 Mobile robot control system

Figure 3.2 Operation of hierarchical system

Figure 3.3 Hierarchical path planning

Figure 3.4 Hierarchical map

Figure 3.5 Image map data

Figure 3.6 Quad tree method

Figure 3.7 Data structure of sub-goal map

Figure 3.8 Relationship between floor plane and image plane

- Figure 3.9** Method to decide avoiding point
- Figure 3.10** Image searching method
- Figure 3.11** Local path planing algorithm
- Figure 3.12** Local map
- Figure 3.13** A method generate avoiding points
- Figure 3.14** Vertical lattice point place point plane
- Figure 3.15** Searching tree
- Figure 3.16** Navigation method
- Figure 3.17** Simulation result 1-(1)
- Figure 3.18** Simulation result 1-(2)
- Figure 3.19** Simulation results 1-(3)
- Figure 3.20** Simulation result 2-(1)
- Figure 3.21** Simulation result 2-(2)
- Figure 3.22** Simulation results 3
- Figure 3.23** Configuration experimental system
- Figure 3.24** Overview of the mobile robot
- Figure 3.25** Mobile robot control system
- Figure 3.26** Experimental result

#### **Chapter 4. センサの協調による移動ロボットのナビゲーション**

- Figure 4.1** Visual following method
- Figure 4.2** Simulation results
- Figure 4.3** Path within one sampling period
- Figure 4.4** Obstacle avoiding path
- Figure 4.5** Simulation results

- Figure 4.6** Sensor data structure
- Figure 4.7** Procedure of sensor information processing
- Figure 4.8** Probability density function
- Figure 4.9** Example of sensor error detection
- Figure 4.10** Error data compensation
- Figure 4.11** Sensor complex method
- Figure 4.12** Sensor integration method
- Figure 4.13** Sensor fusion method
- Figure 4.14** Simulation results
- Figure 4.15** Sensor information fusion
- Figure 4.16** Value of  $w_i$
- Figure 4.17** Simulation result
- Figure 4.18** Production system
- Figure 4.19** Simulation result (before learning)
- Figure 4.20** Simulation result (after learning)
- Figure 4.21** Simulation result for moving obstacles
- Figure 4.22** Experimental mobile robot system
- Figure 4.23** System configuration
- Figure 4.24** Ultra-sonic sensor
- Figure 4.25** Signal data and timing
- Figure 4.26** Overview of mobile robot
- Figure 4.27** Distance data from right-hand ultra-sonic sensor
- Figure 4.28** Experimental result
- Table 4.1** Robot sensor
- Table 4.2** Basic functions of SFS

## Chapter 5. 動的環境における移動ロボットの行動制御

- Figure 5.1 Flow chart of obstacle avoidance method
- Figure 5.2 Feature points of obstacles
- Figure 5.3 Robot coordinate system
- Figure 5.4 Camera system and image
- Figure 5.5 Membership function of relative position
- Figure 5.6 Membership function of static danger
- Figure 5.7 Decision of static danger
- Figure 5.8 Membership function relative velocity
- Figure 5.9 Membership function of dynamic danger
- Figure 5.10 Decision of dynamic danger
- Figure 5.11 Membership function of collision danger
- Figure 5.12 Fuzzy reasoning
- Figure 5.13 Action pattern for obstacle avoidance
- Figure 5.14 Navigation strategy
- Figure 5.15 Simulation results 1-(1) [image processing]
- Figure 5.16 Simulation result 1-(2)
- Figure 5.17 Simulation result 1-(3)
- Figure 5.18 Simulation result 2
- Figure 5.19 Simulation result 3
- Figure 5.20 Simulation result 4
- Figure 5.21 Image data
- Figure 5.22 Simulation result with image noise (2 %)
- Figure 5.23 Simulation result with image noise (5 %)
- Figure 5.24 Simulation results

- Table 5.1** Decision of collision danger
- Table 5.2** Decision table of avoidance behavior
- Table 5.3** Avoidance action patterns
- Table 5.4** Avoidance action
- Table 5.5** An example of action rule

## **Appendix A. マニピュレータのビジュアル制御**

- Figure A.1** Pick and place
- Figure A.2** Visual control
- Figure A.3** Transformation of coordinate system
- Figure A.4** Coordinate system
- Figure A.5** Visual control system
- Figure A.6** Camera image
- Figure A.7** Image data and control signals
- Figure A.8** PDP Network
- Figure A.9** Block diagram of learning process
- Figure A.10** Block diagram of execution process
- Figure A.11** Robotic manipulator with three degrees of freedom
- Figure A.12** Neural Network used for simulation (1)
- Figure A.13** Simulation results
- Figure A.14** Simulation results
- Figure A.15** Simulation model
- Figure A.16** Restrictions to training patterns
- Figure A.17** Simulation results (1) [learned patterns]
- Figure A.18** Simulation results (2) [not learned pattern]

- Figure A.19** Simulation results (3) [not learned pattern]
- Figure A.20** Relation between image and object
- Figure A.21** Visual control system
- Figure A.22** Learning process
- Figure A.23** Neural Network system (Global Network)
- Figure A.24** Neural Network system (Local Network)
- Figure A.25** Execution process
- Figure A.26** Change rule
- Figure A.27** Operating range
- Figure A.28** Global Network
- Figure A.29** Local Network
- Figure A.30** Simulation results (1) [learned patterns]- Global Network -
- Figure A.31** Simulation results (2)
- Figure A.32** Simulation results (3) [not learned pattern]
- Figure A.33** Configuration of experimental system
- Figure A.34** Experimental system
- Figure A.35** Trp-Neuro simulator (AMI)
- Figure A.36** Neural Network (Global Network)
- Figure A.37** Neural Network (Local Network)
- Figure A.38** Experiment for not learned pattern
- Figure A.39** Dynamic servo system
- Figure A.40** Simulation results (1)
- Figure A.41** Simulation result (2)
- Figure A.42** Simulation model
- Figure A.43** Simulation results (for learned pattern)
- Figure A.44** Simulation results (for not learned pattern)



<b>Table A.1</b>	Training patterns for simulation (1)
<b>Table A.2</b>	Training patterns
<b>Table A.3</b>	Pixel error (leaned patterns)
<b>Table A.4</b>	Pixel error (not learned pattern)
<b>Table A.5</b>	Training patterns (Global Neural Network)
<b>Table A.6</b>	Training patterns (Local Neural Network)
<b>Table A.7</b>	Pixel error (learned patterns)
<b>Table A.8</b>	Pixel error (2) (not learned pattern)
<b>Table A.9</b>	Pixel error (3) (not learned pattern)
<b>Table A.10</b>	Simulation results (pixel error)
<b>Table A.11</b>	Experimental results [learned patterns]- Global Network -
<b>Table A.12</b>	Experimental results [not learned pattern]
<b>Table A.13</b>	Pixel error by Neural Network method
<b>Table A.14</b>	Pixel error by Least Square Estimation method
<b>Table A.15</b>	Simulation results

# Chapter 1

## 序 論

本研究の背景、目的について述べる。また、本研究の主要な内容を概観し、本研究の位置づけを示す。

## 1.1 本研究の背景

新しい科学技術を発展させる場として、ロボティクス (Robotics) が最初に脚光を浴びたのは1960年代の後半である。その後数十年にわたる基礎と応用の両方面からの研究の蓄積によって、ロボティクスが飛躍的に発展すると期待されている。その中核をなすのが、マイクロエレクトロニクスの進歩を利用してロボットの知的能力を高める知能ロボット技術である。ここではロボティクスの歴史をふりかえながら本研究の背景について述べる。(Figure 1.1 参照)

ロボティクスという言葉がはじめて使われたのは、技術の世界ではなく、空想の世界 (1950年、アイザック・アシモフ) においてである。ロボティクスとはロボットに関するあらゆる学問・技術体系のことをいう。したがって、横断的な性格をもち、目的達成のためには、全ての知識・技術を利用することになる。ロボットの語源は、チェコ語で労働を意味する "ROBOTA" で、1920年にチェコスロバキアの劇作家カレル・チャペックが戯曲「ロッサム・ユニバーサル・ロボット」の中で使ったのが最初である。当時は、ひとのために働く自動機械を意味し、それは限られた定形的な仕事を繰り返し実行するにすぎなかった。それに対してロボットは、人間が与える多種多様な仕事を、いろいろな環境のもとで独力で実行することが望まれる。この目標に対するアプローチが知能ロボットの研究であり、ある意味では人工知能の基本問題を解くアイデアを試す場としてロボットに挑戦しているとも考えられる。知能ロボットの研究は、1960年代後半から人工知能研究の一つの柱として発展してきた。

このようにロボティクスは人工知能と深いかかわりをもっているが、知識システムのような立場からの研究とは異なった立場をとっているように思われる。すなわち、多くの人工知能の研究は、知識の表現や獲得、記号化された情報に基づく推論などに重点がおかれ、おもに記号世界を取り扱っているといえる。それに対して、ロボティクスはむしろロボットが働く現実の世界と記号世界の接点を取り扱っている。どのように環境のモデルを構築するか、センサ信号の処理の仕方、作業知識の表現方法、作業計画の方法、高性能の手足の制御方法など多くの課題がある。したがって、ロボティクスの世界では、現実の世界やハードウェアに基づいた知的能力の追求が必要になる。

知能ロボットの研究は、1970年代になると半導体・コンピュータ技術の急速な進歩に伴い、マイクロエレクトロニクス技術を結合したロボティクスの進展がめざましくなった。視覚センサや触覚センサを利用して、組立や検査といった複雑な作業が可能な

産業用ロボットがつぎつぎと実用化されていった。1980年代には、いままでの要素技術の蓄積を生かし、より高度なシステムを構築しようという動きが現れる。また、ロボットが使われる分野も拡がり、高精度高速動作を行うロボットばかりではなく、環境認識能力や行動計画能力を備えたロボットが求められるようになってきた。こうして人間に近いロボット、いわゆる知能ロボットの研究が盛んになる。対象となるロボットもハンドアイシステムなどのセンサベースシステムや自律型移動ロボットなどが中心に研究されている。

ロボットが知的な動作をするためには、広範囲な環境に対して適応する能力をもつことが必要である。したがって、ロボットはまわりの状況を知るという環境認識能力をもつことが重要であり、また自分自身が移動できるという能力をもつことも必要不可欠である。ロボットは移動能力をもつことにより、作業空間が拡大し、自ら作業環境内を自由に動いて作業を行うことができる。移動ロボットが自律走行を実現するためには、人間が与えた知識やセンサによって獲得した情報を総合的に判断し、行動を決定しなければならない。そのためには複数の情報を合理的に処理し、目的に応じて行動を律する戦略が必要となる。その際、多種類のセンサを利用して量的ならびに質的に高いレベルの情報を引き出すことが必要であり、センサ情報の処理系が重要な課題となる。

知能ロボットにおいて、各種センサ情報の中で視覚情報は必要不可欠なものであり、多くの研究が行われている。これらの研究の多くは、視覚情報を用いて環境認識を行うものであり、いかにして視覚情報から環境情報を抽出するかに研究の主眼がある。しかしながら、視覚情報は、環境認識ばかりではなく、行動計画および制御においても重要な情報になる。そこで、視覚センサを中心としたロボットコントロールシステムを構成することにより、移動ロボットの行動制御という観点からセンサ情報の利用および処理法を考察する。

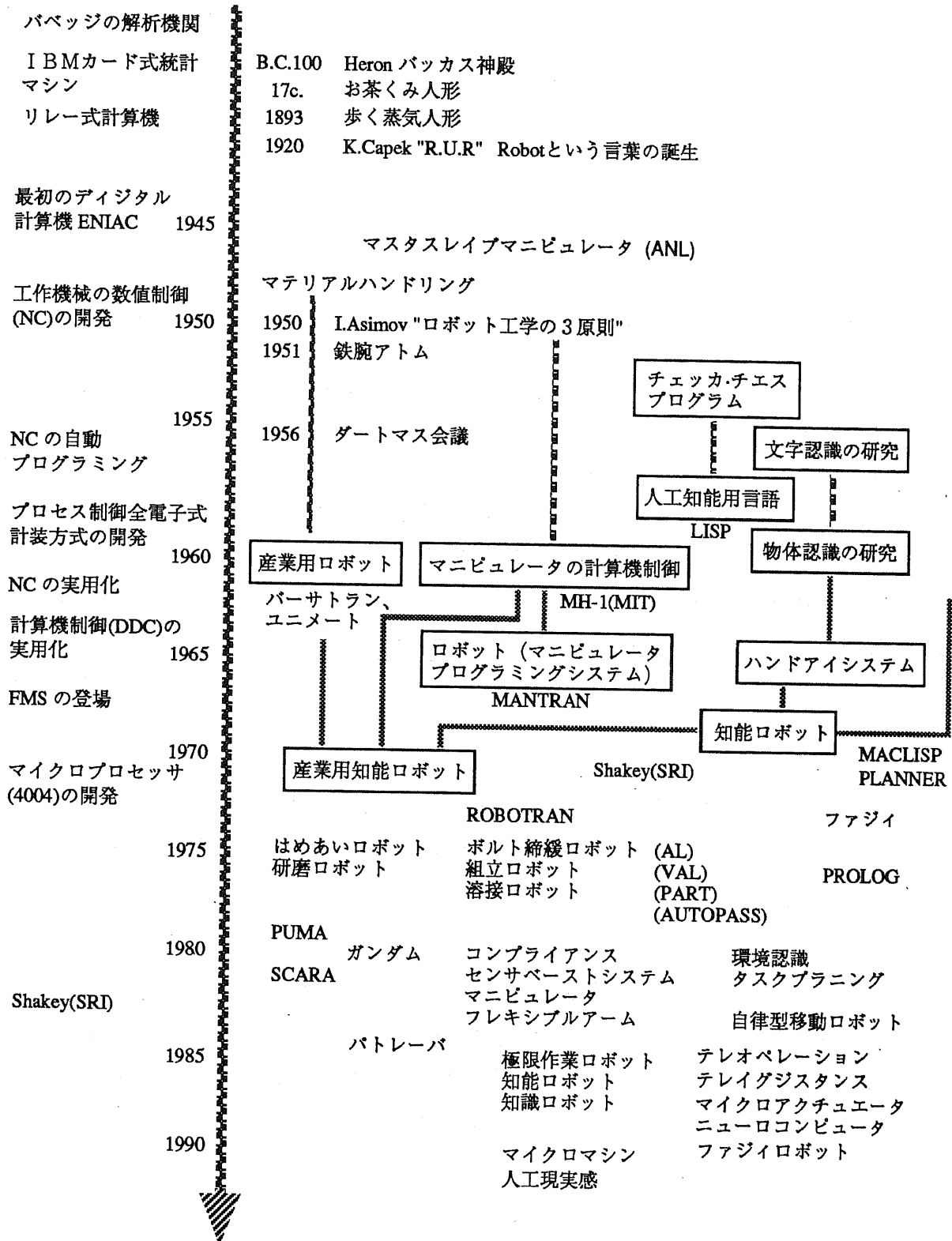


Figure 1.1 History of robotics

## 1.2 本研究の目的

産業用ロボットの利用範囲が広がるにつれて、環境の変化に順応できるような知能をもったロボットが必要となっている。ロボットが知的な動作をするためには、広範囲な環境に対して適応する能力をもつことが必要である。したがって、ロボットはまわりの状況を知るという環境認識能力をもつことが重要であり、また自分自身が移動できるという能力をもつことも必要不可欠である。ロボットは移動能力をもつことにより、作業空間が拡大し、自ら作業環境内を自由に動いて作業を行うことができる。

移動ロボットが自律走行を実現するためには、人間が与えた知識やセンサによって獲得した情報を総合的に判断し、行動を決定しなければならない。そのためには複数の情報を合理的に処理し、目的に応じて行動を律する戦略が必要となる。その際、多種類のセンサを利用して量的ならびに質的に高いレベルの情報を引き出すことが必要であり、センサ情報の処理系が重要な課題となっている。またロボットをコントロールする場合に問題となることは、センシングによって得られた情報をいかに行動計画、制御にフィードバックするかである。従来の方法では、エンコーダ等の内界センサを利用するものが多かったが、ロボットの作業環境の拡大とともに視覚センサ、触覚センサなどの外界センサが使われるようになってきた。特にロボットシステムにおいて視覚センサの果たす役割は非常に大きいものと期待されている。しかしながら視覚情報を画像解析した上で、ロボットを制御する状態量に変換する方式が主に用いられており、演算処理に時間がかかるなどの問題がある。

そこで、本研究は

実時間処理を念頭に、移動ロボットシステムにおいてセンサから得られる情報と運動系の制御情報の整理・統合を行い、センサ情報処理系と行動制御系との有機的な結合を実現することによって、センサベースロボットコントロールシステムを構成すること

を大目的とする。

本研究では、次の3つのフェーズに注目し、本研究の目的であるセンサベースト移動ロボットシステムの実現のための課題を設定する。

### [Phase\_1] 移動ロボットの行動計画

移動ロボットの経路探索問題を取り扱い、地図情報とセンサ情報を有効利用する階層型パスプランニング法を確立する。

### [Phase\_2] 移動ロボットの制御

移動ロボットのナビゲーション問題を取り扱い、視覚センサと超音波センサとの協調を行い、センサ情報融合手法を確立する。

### [Phase\_3] 移動ロボットの行動決定

移動ロボットの動的環境下における衝突回避問題を取り扱い、ファジィ推論によって柔軟な回避行動を実現する。

### 1.3 本研究の概要

本研究では、主に視覚情報に基づいたロボットの行動計画及び制御手法を構築する。ここで取り扱うロボットの作業環境としては屋内環境を想定する。したがって、ほぼ水平な床と垂直な壁に囲まれ、その中に置かれているものも机や本棚といった人工物が多い。また、自然環境に比べてイメージングの条件が良いという特徴を持っている。ここでは、移動ロボットシステムを対象とし、前述の3つのフェーズに応じてセンサ情報処理系と行動制御系との有機的な結合を行う。

まず、Phase\_1において移動ロボットの経路探索問題を取り扱う。経路探索問題に関しては、多くの研究がなされており、さまざまなアルゴリズムが問題の状況に応じて提案されている。本論文では、環境に関する知識が不完全な場合の経路探索法として、センサとして視覚センサを用いた効率的な階層的プランニングシステムを提案する。このシステムは、地図画像情報からサブゴールマップを生成するグローバルプランニングと実際に単眼視による環境観測をおこない、回避ポイントを自動生成しながら移動するローカルプランニングからなる。

次に、Phase\_2において移動ロボットのビジュアルナビゲーションを考える。環境が複雑になるにつれて単眼視による環境観測にはあいまいさが増大する。そこで、このあいまいさを減らすために異種センサを併用し、センサの協調を行う。またこの手法をセンサの融合として位置づける。

そして、Phase\_3において移動ロボットの作業環境内に複数の移動障害物が存在する動的環境を取り扱う。移動障害物を回避するために、まず視覚センサを用いて移動物体のおおよその運動状態を推定し、ファジィ推論によって評価された危険度をもとに回避行動を形成する。本手法を用いることにより人間に近い状況判断能力をロボットに与え、柔軟な回避行動を実現することができる。

さらに、マニピュレータの位置・姿勢制御問題を考える。これは、本研究が対象とする移動ロボットシステムの範中には入らないが、視覚系と制御系との密な結合を実現するという観点から、また移動ロボットシステムにも将来的にはマニピュレーション能力を付加する必要性から問題設定を行う。本研究内容及び得られた知見に関しては本論文の付録にまとめられている。



## 1.4 本論文の構成

本論文は、Figure 1.2 に示すように次の6章からなる。以下に各章の内容を簡単に記す。

### 第1章：序論

本研究の背景、目的について述べる。また、本研究の主要な内容を概観し、本研究の位置づけを示す。

### 第2章：ロボットとセンサ情報

現在実用化されている産業用ロボットの問題点を挙げ、知能ロボットの定義を明らかにする。そして、知能ロボットがもつべき基本的な機能を述べ、さらに知能ロボットにおける情報処理について触れる。次に従来の移動ロボットシステムの問題点を挙げ、自律型移動ロボットシステム実現のための課題について述べる。また、移動ロボットの経路探索、ビジョンベースシステム、センサ融合システムに関して、研究の現状と課題について概観する。

### 第3章：視覚情報を用いた移動ロボットの経路探索

移動ロボットにおいて、スタート地点から障害物を回避しながら目標地点に到達する経路を自動的に計画することは、ロボットがそなえるべき重要な機能である。ここでは、センサとして視覚センサを用い、階層的プランニングを行うことによって経路探索を行う移動ロボットコントロールシステムについて述べる。まず、移動ロボットシステムの構成について述べ、移動ロボットの重要な課題である経路探索を効率よく行うための階層的プランニングシステムを提案する。このシステムの中で、サブゴールマップ、単眼視による環境観測、視覚センサを用いたローカルプランニング、回避ポイントの自動生成法、ナビゲーション法などについて詳しく述べる。シミュレーションによってその有効性を示す。さらに、実験用移動ロボットを用いて、複数障害物回避実験を行い、ローカルパスプランニング手法の実験的検証を行う。

#### 第4章：センサの協調による移動ロボットのナビゲーション

単眼視による環境観測を行い、視覚情報を用いた移動ロボットのナビゲーション法について検討する。単眼視による環境の理解は、必ずあいまいさを伴う。そこで、このあいまいさを減少させるために、超音波センサ等の近接センサを同時に用いる。ここでは、ロボットが必要な情報を信頼性高く獲得するためにセンサ情報の融合を試みる。センサ情報処理として、センサ情報のデータ構造、処理手順について述べる。実際に視覚センサ情報とレンジセンサ情報の融合方法を提案し、シミュレーションによってその有効性を示す。さらに、実験用移動ロボットを用いて、視覚センサと超音波センサの協調による実験を行い、本手法の実験的検証を行う。

#### 第5章：動的環境における移動ロボットの行動制御

複数の移動障害物が存在する環境（動的環境）を対象とし、視覚情報を用いた移動ロボットの障害物回避方法について述べる。まず、視覚センサから得られた画像データを処理することにより、移動物体に対するおおまかな位置及び速度情報を得る。次にこれらの情報を基にファジィ推論を用いて衝突の危険度を評価する。最後にプロダクションルールにより回避行動が自動生成される。本手法の有効性はシミュレーションによって示される。

#### 第6章：結論

本研究のまとめと今後の課題について述べる。

#### 付録：マニピュレータのビジュアル制御

視覚情報を直接利用することによって、マニピュレータの先端を目標位置及び目標姿勢に移動させるビジュアルコントロール法について検討を行う。画像特徴量としては、対象物の特徴点4点に着目する。その4点があらかじめ与えられた目標値に一致するようにマニピュレータへの制御入力を生成する。ここでは、幾何学的な座標変換を行わずに、並列性と柔軟性に富んだニューラルネットワークを用いることによって、画像データとマニピュレータの関節角情報との有機的な結合を行う。また、6自由度マニピュレータを対象としてシミュレーションを行い、提案する制御方式の有効性について検討を行う。また、実験システム（5自由度マニピュレータ）を用いて実験的検討を行う。

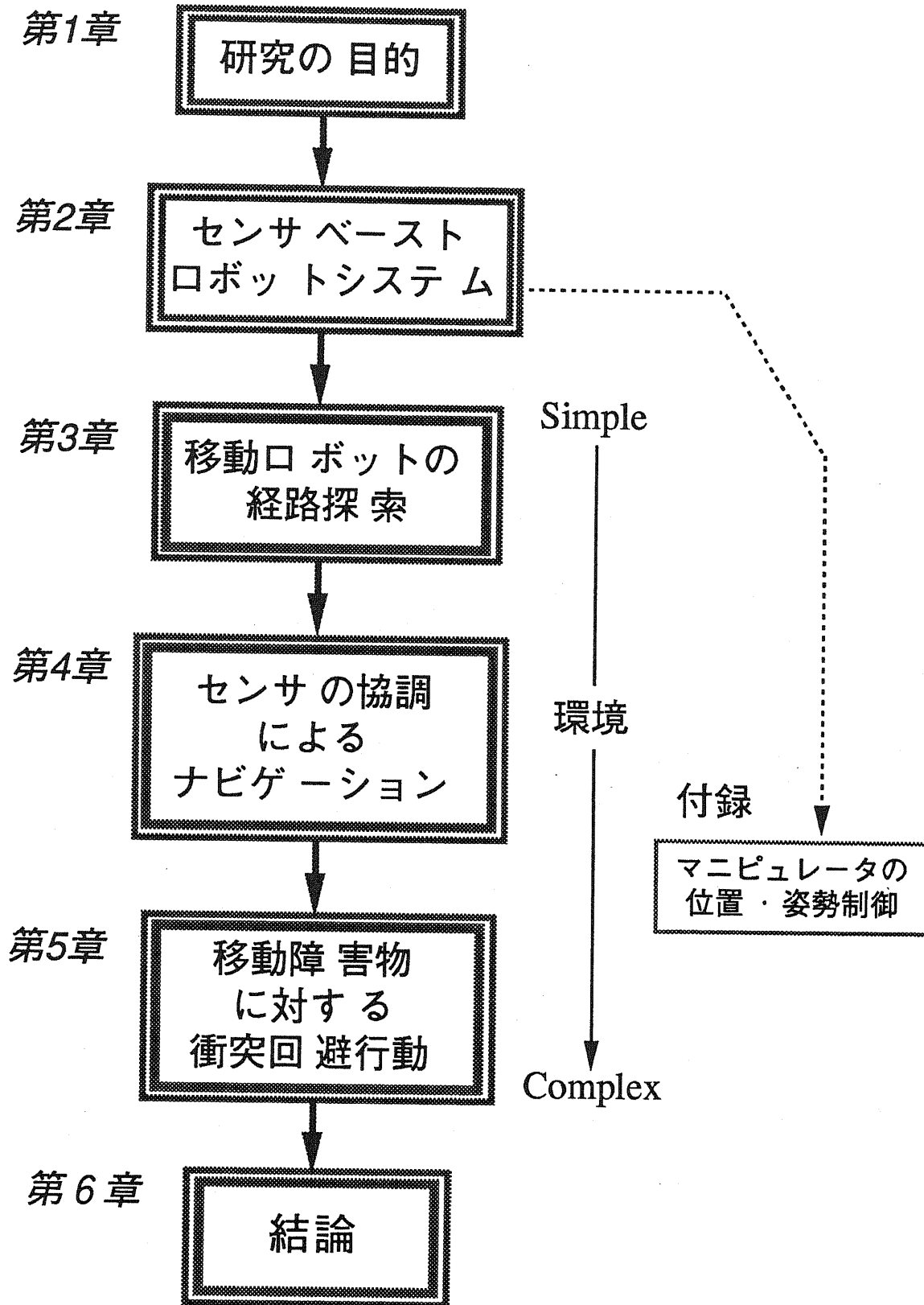


Figure 1.2 Contents

## Chapter 2

### ロボットとセンサ情報

本章では、現在実用化されている産業用ロボットの問題点を挙げ、知能ロボットの定義を明らかにする。そして、知能ロボットがもつべき基本的な機能を述べ、さらに知能ロボットにおける情報処理について触れる。次に従来の移動ロボットシステムの問題点を挙げ、自律型移動ロボットシステム実現のための課題について述べる。また、移動ロボットの経路探索、ビジョンベースシステム、センサ融合システムに関して、研究の現状と課題について概観する。

## 2.1 はじめに

産業用ロボットの利用範囲が広がるにつれて、自ら環境認識機能をもち、環境の変化に順応できるような知能をもったロボットが必要となっている。知能ロボットの研究は、視覚や問題解決など人工知能の諸問題を統合するテストベッドとして取り上げられたものであり、人工知能と密接なつながりがある。ロボットが人工知能の他の分野と大きく異なる点は、実世界との関わりあいの強さである。すなわち、ロボットはセンサによって実世界から情報を集め、行動によって実世界に働きかける。しかも情報の入力、知的情報処理、行動という活動が必ずしも順番に起こるのではなく、必要に応じて適切な活動が選択される。知能ロボット、特に自律型移動ロボットは次世代ロボットの中心的研究課題となると見込まれている。そこでは、センシング技術の融合、環境世界のモデリング、動的な行動計画、知的な問題解決などが要請される。これらを実現する上で知識処理が重要な役割を果たすであろう。

本章では、現在実用化されている産業用ロボットの問題点を挙げ、知能ロボットの定義を明らかにする。そして、知能ロボットがもつべき基本的な機能を述べ、さらに知能ロボットにおける情報処理について触れる。次に従来の移動ロボットシステムの問題点を挙げ、自律型移動ロボットシステム実現のための課題について述べる。また、移動ロボットの経路探索、ビジョンベースシステム、センサ融合システムに関して、研究の現状と課題について概観する。

## 2.2 ロボットシステム

### 2.2.1 産業用ロボットの問題点

ロボティクスの進歩はめざましく、次々と新しいアイデア、素材、技法に基づく研究開発が報告されている [Rheg][Saridis85][Dubourque86][ahmad86]。その中で、産業用ロボットはさまざまな分野で実用化され、生産性の向上などに非常に大きな貢献をしている。しかし、産業用ロボットの利用範囲が広がるにつれて、最近では産業用ロボットの知能化が叫ばれるようになってきた。

従来の産業用ロボットの問題点としては、まずロボットのほとんどがプレイバック方式で用いられており、あらかじめ教示された作業しかできないということである。プレイバックロボットがうまく作業できるのは、作業プログラムを教示する人の頭脳が難しい知的能力を要する部分を代行するからである。従って比較的単純な作業についてはロボットは数多く導入されているが、組み立て等の作業が複雑になる工程にはあまり使われていない。その結果、人間はより複雑で危険を伴う作業を受け持つことになる。これを解決するためには、作業レベルのロボット言語などでロボットに教示できるようにすることが必要で、そのためにはタスクプランニング機能をそなえていることが必要となってくる。

つぎに、ロボットが環境の変化に対して順応できないということも問題点の1つである。従来の産業用ロボットは、工場でのオートメーションの担い手として開発されたものであるため、自動機械の汎用化された程度のもものが多く、かつ人間の手の運動機能の代行という色彩が強く、感覚機能をもっているものは少ない。そのため、作業対象が変わると、対応することができなくなってしまうのである。したがって、ロボットは環境認識機能をもち、環境の変化に対しても順応できるようにしなければならない。さらに宇宙、海底、原子炉内等の人間が長時間作業を行えない環境下では自ら判断し、実行するような知能をもったロボットが必要である。

このように今後のロボットの適用分野を考えるとロボットの知能化はますます必要になってくると思われる。ロボットに高度の知能化を実現し、それを利用するためには外界を認識するための高度の感覚機能が必要となる。人間の場合と同様に多種類の感覚器官を備えていなければ正しく外界を認識することはできない。運動器官や思考部分も同様でこれらを系統立てて整理し、結合したロボットシステムの開発 [Leahy86][Luo86]

が必要になると思われる。

### 2.2.2 ロボットの知能化とは

いままでに『知能ロボット』、『ロボットの知能化』という言葉を使ってきたが、とここでこの『知能』とはいったい何をさすのであろうか、あるいは何ができれば『知能』をもっているといえるのだろうか。これに対する答えはさまざまであり、「外界に対する適応能力」、「環境や条件に従って人間に好ましいように働く機能」などがある。また、JIS用語では知能ロボットを「感覚機能と認識機能によって行動決定のできるロボット」と定義しているが、その答えを一義的に定めることは難しいように思われる。

知能とは、国語辞典によると「物事を見分け、ただしく処理する知恵の働き」とあり、一般にここでは、『知識+判断力』と考えることができる。この知識と判断力がより広く、より深くなれば知能のレベルが高いことになり、それがあるレベルに達すると「知能がある」といわれる。したがって、知能というものは相対的なものであり、時代とともに知能という言葉に要求されるレベルは変化することになる。また、知能はある特定の目的に対して役に立つのであり、目的（行動）の種類によって有意義な知能の種類が異なってくる。

ここでは、ロボットにおける知能を少しあいまいな表現ではあるが、つぎのように考える。すなわち、ロボットの動作が人間からみて知能があると思えること、ロボットの行動が人間にとって知的だと思えることがロボットの知能と考える。このような機能を実現するためには、ロボットは環境を理解するための感覚機能（センサ）を必要とし、また高レベルの情報処理機能をもたなければならない。次節で知能ロボットがもつべき基本機能について簡単に述べる。

### 2.2.3 知能ロボットの基本機能

知能ロボットの機能として行動機能、感覚機能、思考機能がある[若松 a]。このほか、人間との情報のやりとりを行うためのマンロボットインタフェース機能があり、感覚、行動、思考の諸機能の足りない点を人間が補佐したり、ロボットの内部状態を人間に知らせたりする場合に使われる。

### 行動機能

外界に対して働きかける機能であり、人間の手足に相当する。手は対象物に対して作用し、実際にそれを操作するもので、いわゆるマニピュレータであり、多関節形、伸縮するもの、柔軟なもの、多自由度の手先をもつものなど、種々の形式のものがある。足は移動に用いるものであり、車輪やキャタピラの他、歩行、蛇行、吸盤式のものなどがある。

### 感覚機能

自分自身の姿勢など、体の状態を知る内界センサと外界の情報を取り込む人間の五感に相当する外界センサがある。内界センサはマニピュレータや移動機構などの運動機構の内的状態を検出するものである。マニピュレータの各自由度の変化量や変位速度を検出するためのエンコーダ、ポテンシオメータ、タコジェネレータなどがこの例である。外界センサはロボットの周囲状況を知覚するためのもので、視覚、聴覚、臭覚、味覚、触覚という人間の五感に相当する。

### 思考機能

外界に関する情報を用いて、与えられた仕事を実行するための計画をたてる機能で、人間の脳に相当する。プログラムを記憶しプランをたてたり、画像の特徴による物体識別などのパターン認識の機能を実現し、高度な判別を含む作業のプログラムも可能である。また、状況に応じた作業計画の立案も可能である。さらに、手や足を動かす運動神経系の機能、それらの協調管理の機能も有する。

### マンロボットインタフェース

人間とロボットとの間の情報交換にはロボットから人間に対するものと、人間からロボットに対するものとが考えられる。前者は、ロボット自体や周囲状況に関する情報を人間に伝達するもので、通信と表示技術によるところが大きく、後者はロボットに対する環境教示手法、作業の記述を容易にするためのロボット言語、人間の意志の最適な伝達手段である自然言語と音声の導入などがある。



### 2.2.4 ロボットのセンサ制御系

一般的なセンサ制御システムは Figure 2.1 に示すように次の5つのレベルに分類できる [増田 87]。

1. シーケンス制御レベル
2. フィードフォワード制御レベル
3. センサフィードバック制御レベル
4. センサ適応制御レベル
5. センサ自律制御レベル

1. のシーケンス制御レベルは、オンオフセンサの順序制御・条件制御でロボットとマシンとのインタロック制御や協調同期制御がこのレベルである。2. のフィードフォワード制御レベルは、情報収集、認識後の動作制御・オープンループ制御である。3. のセンサフィードバック制御レベルでは、センサ信号による演算フィードバック制御であり、現状の感覚制御の主流である。例えば、圧覚センサによる握力制御の際の PID 演算フィードバック法がこのレベルにあたる。4. のセンサ適応レベルでは、感覚センサ情報が直接制御動作に対し扱われるのではなく、最適性の追求と状態の適応のため制御システムのパラメータ調整が行われる。5. のセンサ自律制御はロボットの感覚制御系の最終目的であり、判断、推論、意思決定などの AI 機能が要求される感覚制御系である

### 2.2.5 ロボットシステムにおける情報処理

ここで情報処理の立場からロボットをながめてみることにする。ロボットは、前節で述べたように運動部、感覚部、思考・判断部、マンロボットインターフェース部からなり、相互に密接な関連をもちながら機能している。たとえば、効果器の場合、最適の行動規範に従って動くためには、最適制御や適応制御などの高度な制御アルゴリズムを利用しなければならないし、組立作業などの場合には、感覚器からの情報に基づく複雑な制御が必要である。また、視覚は頭脳の一部といわれているように、高度な思考・判断を必要とする場合が多い。ここでいう思考判断部とは、作業対象や作業

環境を観察してその状況を認識したり、人間の意図を理解して行動決定をしたりするような、人間における脳の働きに似た、レベルの高い思考活動をさすものとする。

ロボットにおいてはさまざまな場面で情報処理が行われているが、その代表的なものとして視覚情報処理について簡単に触れてみる。人間は、感覚器官から得られる情報の80%以上を視覚から得ているといわれており、ロボットにおいても環境を認識する場合、いろいろなセンサの中で視覚センサの役割は大きなものであると考えられる。TVカメラなどの入力装置を介して取り込まれた画像入力に基づいての3次元の情景判断は、人間の眼と比較して情報量が圧倒的に少なく、しかもその入力はさまざまな雑音を含んでいる。その中から必要な特徴量を抽出したり、対象物を識別したり、状況を理解したりする試みは、コンピュータビジョンとよばれ、ロボットの視覚もこの分野に含まれている。特にロボットの視覚に限定したコンピュータビジョンはロボットビジョンとよばれ、盛んに研究が行われている。このほか、ロボットは情報処理のさまざまな分野と関係しており、状況判断、推論、問題解決、学習、自然言語理解など一般に、知識情報処理、知識工学あるいは人工知能といわれる情報処理の1部門と密接な関係をもっている [Rich]。

近年のマイクロエレクトロニクスの発達、人工知能研究の活発化、知識情報処理分野の研究成果により、ロボティクスの研究も産業用ロボットの研究から、しだいにより柔軟で広範な作業のできる知能ロボットの研究、さらには問題解決、学習能力を有する知識ロボットへと研究の主眼が移りつつある [辻86][井上90]。研究対象もマニピュレータのほかにセンサベースシステム、移動ロボット、フレキシブルアーム、自由度の大きい指などにひろがってきている。

本研究では、移動ロボットシステムを対象とし、センサ情報処理、環境認識、行動計画などの種々の機能を統合化することによってセンサベース移動ロボットシステムの構成を考えることにする。

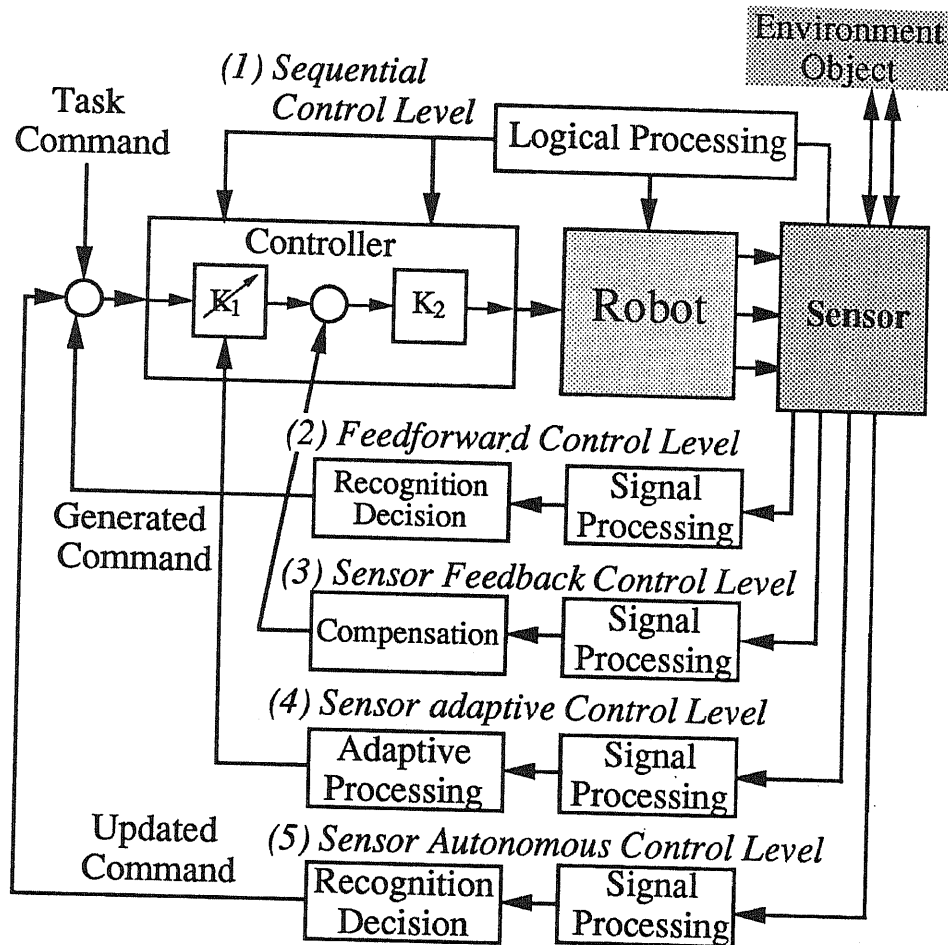


Figure 2.1 Sensor control level

## 2.3 移動ロボットシステム

ロボティクスの進歩はめざましく、特にマイクロエレクトロニクス技術の成果を活用したロボットの知能化が注目を浴びている [油田 89]。ロボットが知的な動作をするためには、広範囲な環境に対して適応する能力をもつことが必要である。したがって、ロボットはまわりの状況を知るという環境認識能力をもつことが重要であり、また自分自身が移動できるという能力をもつことも必要不可欠である。

### 2.3.1 自律型移動ロボット

ロボットは移動能力をもつことにより、作業空間が拡大し、自ら作業環境内を自由に動いて作業を行うことができる。また、認識への貢献もあげられる。いままでの視点を固定した1枚の画像のみを対象とする認識には、それなりの不確実性が本質的に残ることは否めない。認識装置が自分自身の判断に基づいて自由な移動を行い、視点を認識する側の主体性によって自由に変えて3次元認識を積極的に行う方式が、認識の問題の解決に貢献する可能性は十分にあると考えられる。ここが、移動ロボットの視覚とコンピュータビジョンの研究の大きな差でもある。コンピュータビジョンは、与えられた画像の解釈を受動的に行うのに対して、ロボットは自分の環境のモデルを総合的に、かつ能動的に作成するからである。

ところで、ロボットが移動を行い、行動の自由度が増すと、それと共に要求される知能の種類も大幅に広がってくる。自律型移動ロボットに要請される主たる情報処理プロセスとしては、つぎのようなものが考えられる [小林 87]。

#### 【情報処理プロセス】

1. 人間から与えられた命令を理解するプロセス
2. センサから獲得した情報を総合的に解析、解釈して環境を理解するプロセス
3. 目的を達成するために、実行可能な行動系列を計画するプロセス
4. その行動計画をアクチュエータへの制御指令に展開するプロセス

ここで、ロボットが『自律的』であるとは

- 抽象レベルで与えられた命令を実行可能な計画として自ら構成できる能力をもつこと
- センサによる環境情報の収集と分析を能動的に行う能力をもつこと
- そして、得られた情報・知識から自ら行動決定できる能力をもつこと

をいう。したがって、ロボットの自律性は外部からのエネルギーや情報の供給をいっさいたつというのではなく、むしろどのくらいの時間、外からの助けなしで独立で1つの目的のために行動することができるかがポイントとなる。

### 2.3.2 従来の移動ロボットシステムの問題点

移動ロボットのコントロールシステムには、おもにセンシング、プランニング、コントロールの機能[若松b]がある。移動ロボットをコントロールする場合に問題となることは、センシングによって得られた情報をいかに行動計画、制御にフィードバックするかである。従来の方法では、移動ロボットは静止座標系によって表された環境地図に基づいて制御されている。一方、環境はロボットに搭載されたセンサによってセンシングされるので、センサ情報をロボット座標系から静止座標系に変換する必要があった。ここで、この変換を行うには移動ロボットが静止座標と対応づけられる基準点からどれだけ移動したかを知る必要がある。この情報は、一般に車輪の回転数をエンコーダ等で検出するといったような内界センサの利用によって得られる。しかし、移動距離を車輪の回転数で測定すると、スリップなどにより誤差が集積されやすい。この誤差を視覚センサなどの外界センサを用いて位置の同定や地図の更新をする研究[Moravec83]も行われている。しかし、一般の環境で、視覚センサなどによって静止座標系でのロボットの位置を決定するためには、画像の処理時間がかなりかかるとともに、非常に多くの知識が必要になってくる。

ロボット座標系から静止座標系への変換が、たとえ正確に行われたとしても、移動ロボットは、静止座標系の環境地図なしで移動することになる。このような方式では、時々刻々得られるセンサ情報を直接フィードバックできないため、動的な移動を扱うことは困難である。そこで、高度な地理理解システムをもち、センサ情報によって、リアルタイムに地図情報を更新するロボットの研究[Wallance86]も行われている。また、さまざまな機能を必要とする移動ロボットシステムにおいて、従来のシステムでは各機能

をただ結合したにすぎなかったが[Nakano85]、移動ロボットにおける各機能の使われ方を考慮して、それらの機能を統合しようという研究[Brooks83][Crowly85][Kriegman87]が徐々に行われてきている。

移動ロボットのシステム構成としては、各サブシステムを階層型に結合していくのが一般的であり[Bhatt87][Montgomery86]、本研究でも階層型のシステム構成を考える。

### 2.3.3 自律型移動ロボット実現のための課題

産業用ロボットに代表される作業ロボットでは、環境が変化することはあまりなく、また繰り返し作業が主であるために環境理解のプロセスはそれほど重要ではない。一方、移動ロボットの場合、動く環境は、工場のようにあらかじめ人工的に設定できるとは限らず、いろいろな環境に適応する行動をとる必要がある。いうまでもなく、センサ情報の処理・運動部の制御は高性能でしかも信頼性が重要になり、ロボットに高次の理解能力および問題解決能力もたせることが不可欠である。

自律型移動ロボットを実現する上で解決すべき基本的な問題としては、つぎのような課題がある。

#### 環境モデルの作成と更新

ロボットが移動する環境をあらかじめ完全にモデル化しておくことは困難であるため、おおまかな環境モデルしか与えることができない。したがって、ロボット自身に必要なに応じて詳細な環境モデルを構築し、これを更新していく能力をもたせなければならない。

#### 各種センサの統合的利用

環境モデルを構築する上で、各種センサ情報をいかに統合するかが問題となる。移動ロボットでは、レンジセンサやソナーなどによる距離情報および単眼視やステレオ視、動画像などによる視覚情報を融合して、環境の理解についてのあいまいさをできるだけ減らして、より完全に近いモデルを構築することが必要である。また、それらを問題解決の抽象レベルに応じて利用できるように、階層的に表現しておくことも重要である。

## 行動計画の決定

一般に移動ロボットでは障害物を回避しながら目標点に到達するための経路探索が要請される。経路探索には、試行錯誤的な側面が伴い、観測によって環境モデルが動的に更新されなければならない。

### 2.3.4 移動ロボットにおける情報処理モデル

いままでみてきたようにロボットに自律性をもたせるためには解決すべき課題が数多く残されている。その中で重要な課題の1つは情報の多様性に起因する問題である。たとえば、移動ロボットが自律走行を実現するためには、人間が与えた知識やセンサによって獲得した情報を総合的に判断し、行動を決定しなければならない。そのためには複数の情報を合理的に処理するモデルおよび目的に応じて行動を律する戦略が必要となる。

移動ロボットが用いる情報は、獲得する手段の違いによって、2つに大別できる。1つは人間から与えられる知識であり、もう1つはロボット自身が感覚機能により得られるセンサ情報である。知識は性質上、「環境に関する情報」と「処理に関する情報」に分けられる。前者は環境の形状や障害物に関する知識であり移動ロボットにおいて環境地図の役割を果たす。処理に関する知識とは環境認識や経路計画などを行うためのルールやアルゴリズムである。一方、センサ情報はセンサの分類上、テレビカメラやレンジセンサなどの外界センサによる情報とエンコーダやポテンシオメータなど内界センサによる情報に分けられる。移動ロボットはこれらのさまざまな知識や情報を総合的に判断して行動しなければならない。

移動ロボットにとってこのようなさまざまな情報や知識を統一的に処理可能な情報処理モデルが必要である。また、環境に関する知識やセンサ情報はあいまいさを含んでおり、モデル自体があいまいな情報を定量的に処理できることが必要となる。さらに環境認識、経路探索、障害物回避などの問題に対して有効なモデルでなければならず、かつ高速な処理アルゴリズムが必要となる。現在提案されているモデルとしては、関数表現によるもの、グリッドモデルなどがあるが、あいまいな情報の定量的な扱いができないなどの問題がある。最近あいまいさを考慮できるモデルとして境界距離モデル [池上 89] も提案されている。上記の問題を合理的に解決し得る移動ロボットのための情報処理モデルが望まれる。

## 2.4 移動ロボットにおける経路探索問題

移動ロボットにおける経路探索問題とは、与えられた環境のもとで初期点から出発して、障害物を回避しながら目標点に到達するような経路を見いだすことである。移動ロボットの経路の探索方法 [柿倉 90] は、環境に関する知識が事前にどれだけ与えられているかに大きく依存するため、

- 環境に関する知識がない場合
- 環境に関する知識が完全な場合
- 環境に関する知識が不完全な場合

の3つの場合に大別して考えてみる。

### 2.4.1 環境に関する知識がない場合の経路探索

この経路探索の状況は未知の地域へ地図なしで行った人間の探索状況と似たものである。すなわち、経路を探索するロボットは、出発点において適当なセンサを用いて周囲を観測し、目標点を探す。観測しながら移動し、得られた新しい情報に基づいて、より目標点に近いと思われる方向に動くことにより経路の探索を継続する。ロボットはこのような過程を目標点に到達するまで繰り返す。このような方法は試行錯誤的探索法あるいは網羅的探索法であり、一般に後戻りを必要とするが効率の是非はともかく、目標点を見いだすことができる。ただし、最適な経路とは限らない。

### 2.4.2 環境に関する知識が完全な場合の経路探索

目標物の位置および障害物の位置と形状に関する情報などが完全に与えられている場合には、試行錯誤的探索法よりももっと効率的な探索手法（数理計画法探索）が適用できる。ロボットの初期位置および最終位置を与えるだけでそれらを結ぶ障害物回避経路を自動生成するアルゴリズムが多数提案されている [Nilson69][登尾 89]。

探索領域上に適当な代表点を選び、これをネットワークの節点に対応させ、また2節点間が移動可能であるとき、これをリンクに対応させれば、計算幾何学の可視グラフ



問題に帰着させることができる。この考え方を適用すれば、動的計画法や分岐限定法などのネットワーク最適化手法の適用が可能になる。最短経路はネットワーク上で出発点から目標点に到る最短路を求めることに対応し、ラベリング法などによって、可視グラフの節点数を  $n$  とすると  $O(n^2)$  で計算できることが知られている。コンフィギュレーション空間、一般化錘体などに基づいたグラフ探索法では、グラフ上に障害物回避動作が含まれるときには必ずそれは獲得できる。しかし、グラフ探索法には、グラフを作成するための前処理、例えば作業空間の障害物領域を侵入禁止領域に成長させることでロボットを点に帰着させる、*Growing Obstacle* という処理を行う必要があるが、これには膨大な計算量が必要となる。また、ロボットや障害物の形状が複雑になればなるほどグラフのノード数が膨大になり、経路計画にもかなりの計算時間がかかるなどの問題がある。

一方、ポテンシャル法には、人工ポテンシャル法 [宮崎 85][Khatib86]、楯円ポテンシャル法 [奥富 83]、ラプラスポテンシャル法 [佐藤 87b] などがある。人工ポテンシャル法と楯円ポテンシャル法は目標点への吸引力と障害物からの反発力のつりあいによりロボットの動作を生成するので、吸引力と反発力がつりあって、動作が生成できなくなり最終位置以外で静止することがある。また、ロボットと障害物が干渉する可能性がある。さらに、円や長方形など簡単な形状で表現しなければならないなどの問題がある。ラプラスポテンシャル法はラプラス方程式を解いて作業空間のポテンシャル場を作成するのに膨大な時間を必要とし、障害物の配置や最終位置の変化に柔軟に対応できない。

このグラフ理論あるいは数理計画法に基づく経路探索は、理論的には必ず最適解が求まるが、実際には問題が複雑になると組合せの数が爆発的に膨大になってしまうなどの問題が生じ、何らかの工夫が必要となる。

### 2.4.3 環境に関する知識が不完全な場合の経路探索

環境に関する知識が部分的にしか与えられていない場合には、何らかの発見的方や、いくつかの手法を組み合わせるハイブリッド法が有効である。発見的手法に用いられる発見的知識としては次のようなものが考えられる [柿倉 87]。

#### 発見的知識

1. 現在地点から目標地点へ向かって、障害物に遭遇するまでまっすぐ進む。
2. 障害物に遭遇したら、現在地より見える障害物の端点で最も近いものところまで行く。
3. 障害物に沿って移動する。

知られている知識だけから全体の経路計画を作成したとしても、未知の障害物の存在により、計画が実行不可能になるおそれがある。このことは計画したときにはわからないので、障害物が発見された時点で、計画を変更するなどの手続きをする必要が生じる。

## 2.5 ビジョンベースロボットシステム

ビジュアルコントロールシステムの構成は Figure 2.2 に示すように、認識レベル、特徴レベル、センサレベルの3つのレベルからなる。認識レベルに関しては、コンピュータビジョンの分野などで盛んに行われている。センサレベルに関しても、視覚情報を画像解析したうえで、ロボットを制御する状態量に変換してフィードバックする方式が主に用いられきた。しかしながら、画像処理および演算処理に時間がかかるなどの問題がある。これに対して最近では、中間の特徴レベルにて、視覚系とロボット制御系の密な結合を実現するために、視覚特徴量を直接ロボットの制御に結び付ける研究が注目を浴びている [佐藤 87a]。これは、画像上の物体の面積や線分の長さなどの画像特徴量を用いてロボットを直接的に制御し、制御系のダイナミックな特性を向上させようというものである。

### 2.5.1 ロボットシステムにおける視覚情報の役割

知能ロボットにおいて重要な役割を果たすものに、環境認識機能がある。環境認識には大きく分けて2つの段階が考えられる。第一段階は、環境に関する情報の獲得である。ここでは、センサの利用法がポイントになる。第2段階は、センサによって得られた情報の解釈である。センサによって得られる情報は、そのままではタスクプランニングやコントロール等に利用できるものではなく、各種センサ情報を処理・統合化して1つの環境として解釈する必要がある。この2つの段階の内、後者の環境の解釈は、環境に関する大量の知識を利用した適切な推論が行えなければ、すべての環境において正しい解釈を行うことができない。現段階では、あらゆる環境において正しい解釈を行うようなシステムを構成するのは難しく、ある一定の条件によって単純化された環境に対処できるシステムを構成することになる。ロボティクス研究においては、そのような単純化された環境において、第1段階のセンサの利用方法を含めて環境情報の獲得法を研究することが主眼になる。対処できる環境の範囲はマイクロコンピュータ技術や知識情報処理技術の発展に伴って広がっていくと思われる。

環境情報を得るためのセンサとしては、人間の感覚器官に対応するものとして、視覚センサ、聴覚センサ、触覚センサなどがある。人間は、感覚器官から得られる情報の80%以上を視覚から得ているといわれており、ロボットにおいても視覚センサの果たす役割は非常に大きなものになると考えられている。このため、ロボティクス研究

においても、ロボットビジョンは盛んに研究されてきた。ロボット環境に関する情報のなかで、最も重要になってくるのは、ロボットから対象までの距離情報（環境の奥行き情報）であり、ロボットビジョンの研究においても、視覚から環境の奥行き情報を抽出することが研究の中心である。しかしながら、視覚情報において、環境の奥行き情報は基本的に欠落しており、画像から奥行き情報を得るためには、特殊なテクニックが必要になる。このようなテクニックとして、ステレオビジョン、レンジファインダ、オプティカルフローの利用などがあり、さまざまな方法が提案されてきた[辻87]。

環境の奥行き情報は、ロボットがその環境で動作する場合必要な情報であるが、それを視覚センサだけから得ようとするのは、あまり得策ではない。画像において環境の奥行き情報は欠落しており、視覚センサが得意とする情報ではない。視覚情報の最大の特長は、情報の大域性にあり、これを生かすような視覚センサの利用が望ましい。しかしながら、ロボットビジョンの研究においては、奥行き情報の抽出に集中するあまり、ロボットビジョンから制御情報のような別の情報を得ようとする研究が少ないのが現状である。

### 2.5.2 視覚を用いた移動ロボットシステム

移動ロボットの視覚システムの枠組みは、あらかじめ環境に関して詳しい知識をもっている場合と未知の環境での移動に分けられる。前者の場合には、建物の構造や形状についての詳細な知識を地図の形で与え、ロボットがそれを利用して観測と移動を行う方法が多く研究されている。3次元世界を階層モデルで表現し、イメージモデルと入力画像を比較する方法[辻86]、ロボットが移動する環境における路面の標識などのランドマークを用いる方法[森87]、屋内環境を移動するために地図とカラー画像を利用する方法[坪内87]などが提案されている。一方、後者の場合には、主にステレオ視を行い、3次元世界での物体の認識を行っている。初期の研究で、インタレストオペレータで特徴点を検出し、その対応を相関計算で決定し、ロボットが移動と観測を繰り返して、環境の地図を作成する方法[Moravec83]が提案されている。また、多地点の観測結果をカルマンフィルタを用いて処理し、信頼性の高い世界モデルを作成する方法[Ayache87]も提案されている。

このように、視覚センサを搭載した移動ロボットにおいて走行環境に関する地図を利用することは有効と思われるが、画像処理や対応点の検出（ステレオの場合）に時間がかかるなどの問題があり、実時間処理を行うためにはなんらかの工夫が必要になる。

### 2.5.3 マニピュレータのビジュアル制御

本研究の対象は移動ロボットシステムであるが、さらに自由度の大きい3次元空間を移動するマニピュレータに対しても視覚センサ情報の有効的な利用を考察する必要がある。また、最近、並列性と柔軟性を有するニューラルネットワークのロボット制御への応用が注目されている。そこで、ニューラルネットワークを用いてマニピュレータの先端を目標の位置及び姿勢に移動させるビジュアル制御について検討を行った。研究方法及び結果に関しては本論文の付録にまとめられている。

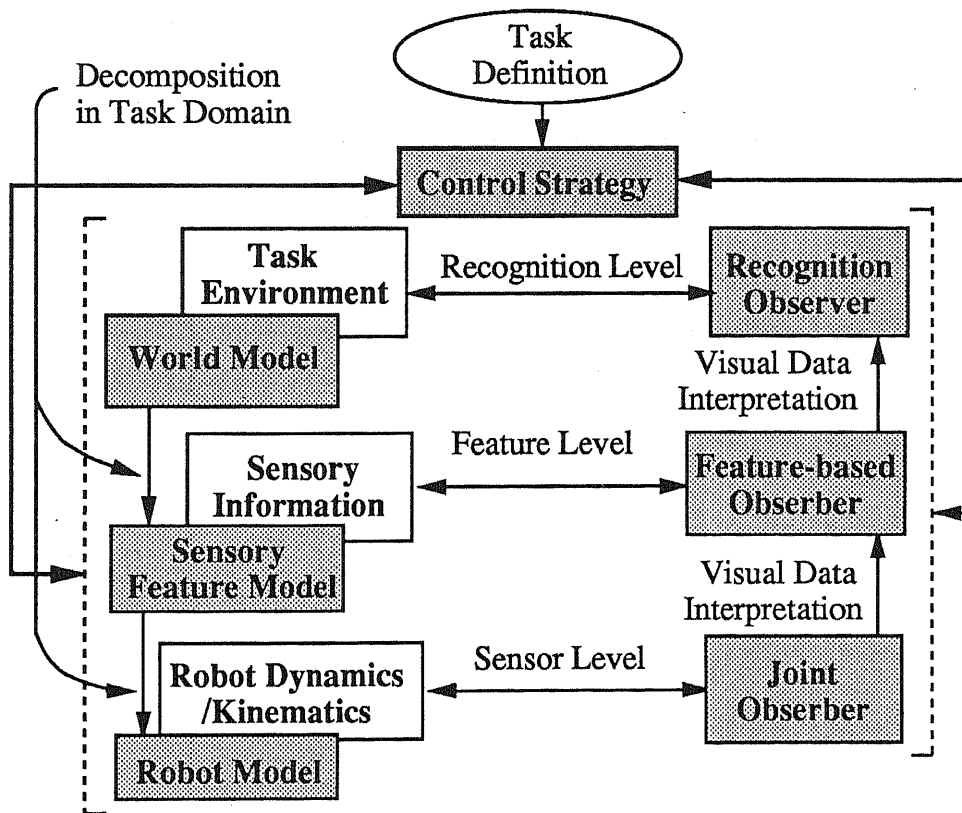


Figure 2.2 Hierarchical structure

## 2.6 センサ融合システム

知能ロボットをはじめ、航空機、自動車、マニファクチャリングなどの分野を中心に、システム全体で用いるセンサの数や種類が増大するにつれて、多種類・大量のセンサ情報に対する効率的な処理システムが必要である。多種類のセンサ情報を扱うということは、それだけ処理すべき情報量の増大を意味し、さらにセンサ間の矛盾を生じやすく、知識の利用も含めてこれらに対処する技術の開発が不可欠である。センサ融合システムの研究は、このような問題の解決をめざすものであるが、研究の歴史が浅いのでその考え方は整理されておらず、指針もまだ明確ではない。そのため、現状では一般的に確立された方法論はなく、将来に多くの課題を残している。ここでは、センサフュージョンの考え方を中心に現状と課題を概観する [Luo89]。

### 2.6.1 センサ情報融合とは

人間は、視覚、聴覚、触覚など数多くの感覚情報を用いて、外界や自己の変化を知覚・認識している。このように多種類の感覚情報から融合された知覚は、視覚のみあるいは触覚のみの情報に比べて確実性が高く、多義的な情報を提供することができる。人間の脳には、このような多種類の感覚情報の処理機構として、ネットワークにより相互に結合された階層的並列分散処理機構が存在するといわれている。

このような処理系をセンサ情報の処理機構として工学的に実現しようとする考え方がセンサ融合である。すなわち、複数のセンサの情報から、単一のセンサのみでは得られない新たな情報を抽出しようとする考え方である (Figure 2.3)。近年になって、ロボット工学や自律システムなどの研究分野でその必要性が指摘され、センサ技術における新たな課題として議論されている [木下 90]。

### 2.6.2 センサフュージョン

複数のセンサから得られる情報処理機構はセンサの種類などに応じて処理内容が異なってくるので、全体を統一的に考えることは難しく、その処理系を総称してセンサフュージョンと呼ばれている。多種類のセンサを用いて、目的とする情報を導出するための処理手法には Table 2.1 に示すように4つの形態が考えられている [石川 90a]。

### (1). センサ複合

「複合」的処理は、複数のセンサからの情報を並列的あるいは相補的に組み合わせた出力を得る過程である。このような形態はごく基本的なもので、出力形態の統一化、センサの集積化、センサの選択といった主としてハードウェアレベルの問題となる。単一機能性・局所性の回避、測定レンジの拡大などが主な目的である。(Figure 2.4 参照)

### (2). センサ統合

センサからの情報になんらかの演算処理を施してまとまった情報を得ようとする場合には、「統合」的処理となる。平均処理や温度補償などがよい例で精度・信頼性の向上、処理時間の短縮などが目的である。(Figure 2.5 参照)

### (3). センサ融合

センサ情報間あるいはセンサ情報と内部モデルとの間で相互の関係が新たな知覚表象を与える場合には「融合」的処理となる。両眼融合や視触覚融合がよい例で、物体認識や空間認知などで利用できる。(Figure 2.6 参照)

### (4). センサ連合

センサ情報に処理を施して何らかの処理結果を得るという一方向性の処理ばかりでなく、センサ情報間の関係を理解する「連合」的処理も必要である。認識の対象、さらには予測や学習・記憶、異常の検出などが主要な目的である。

## 2.6.3 センサフュージョン研究の現状と課題

センサフュージョンを実現するための具体的な課題としては、Figure 2.7 に示すように以下の3つがある。

1. 上層での処理の負荷の軽減と統一化を図るため、機能的に局在した処理機構を効率的に実現するセンサの知能化を中心としたハードウェアの問題

2. 効率のよい分散処理やリアルタイム処理などに対する記述論を主体としたソフトウェアの問題
3. これらの基盤のうえに実現する計測アルゴリズムの問題

このうち、ハードウェアやソフトウェアの問題は、いわば道具の整備の問題であって、中心はアルゴリズムの問題にある。このアルゴリズムの問題には、従来からの信号処理を異種情報に適應することを主眼とした信号処理の問題、知識や記憶なども含めて計測の論理構造を実現しようとする人工知能の応用の問題、計算機構造や論理構造が明示的に得られない場合にはニューラルネットの問題として扱われている。

#### 2.6.4 能動型センサ融合システム

センサ融合では、センサ情報間に共通の基体を設定することが課題であり、この基体がセンサ融合の構造を決定する最大の要因である。一般的には、対象の特徴や内部に仮定したモデルをア・プリオリに共通の基体とすることが多いが、自律性が高い場合には、異種センサ情報、特に相互の独立性が高いセンサ情報間では効力を持たない。そこで、共通の基体を提供するものとして能動的な運動が考えられる。いま、2種類のセンサ  $\phi_1$ 、 $\phi_2$  の機能を対象  $X$  に対して

$$Y_1 = \phi_1(X), \quad Y_2 = \phi_2(X) \quad (2.1)$$

とした場合、この出力  $Y_1, Y_2$  が独立であると、センサ融合は達成されないが、能動的な運動  $V$  を導入することにより、

$$Y_1 = \phi_1(X, V), \quad Y_2 = \phi_2(X, V) \quad (2.2)$$

とすれば、 $V$  は制御可能であるから、その設計により  $Y_1, Y_2$  の独立性を最小にすることができると考えられる。このことは、従来からの「検出」→「処理=認識」→「行動・運動の生成」→「行動・運動」という一方向性の図式を書き直すことになり、認識を単にセンサ情報の処理過程と考えるのではなく、何らかの運動と一体化した能動的プロセスと考えることになる [石川 90b]。



センサフュージョンは特に自律的な機能を必要とする分野において問題意識が高く、センシングシステムの構造を左右する重要な課題である。多種類のセンサを組み合わせることはそれほど簡単ではなく、将来に多くの問題が残されている。

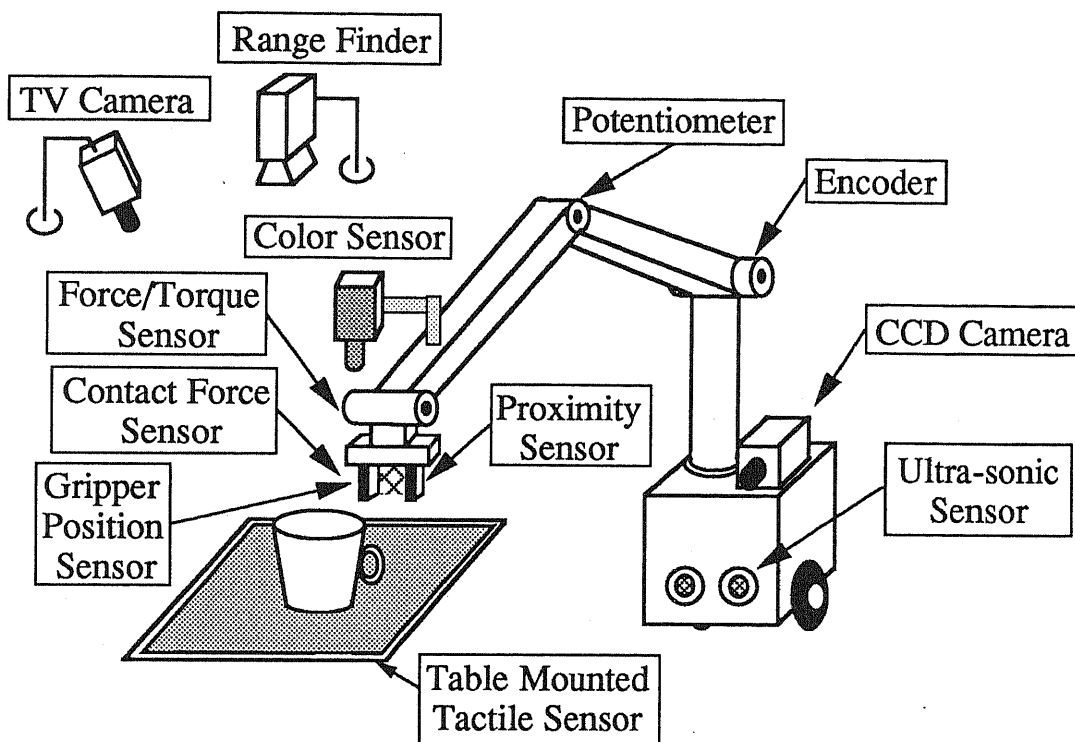


Figure 2.3 Multiple sensor system

Table 2.1 Sensor information processing

types	processing sensor data (A,B)
sensor complex	$A, B \Rightarrow A + B$
sensor integration	$A, B \Rightarrow f(A + B)$
sensor fusion	$A, B \Rightarrow C$
sensor association	$A, B \Rightarrow (A \rightarrow B, B \rightarrow A)$

# Sensor Complex

Operation: additive or complement processing

Effect: avoidance of locality or single functionality  
expansion of measurement range

Range finder

→ 3D distance color image

Color camera

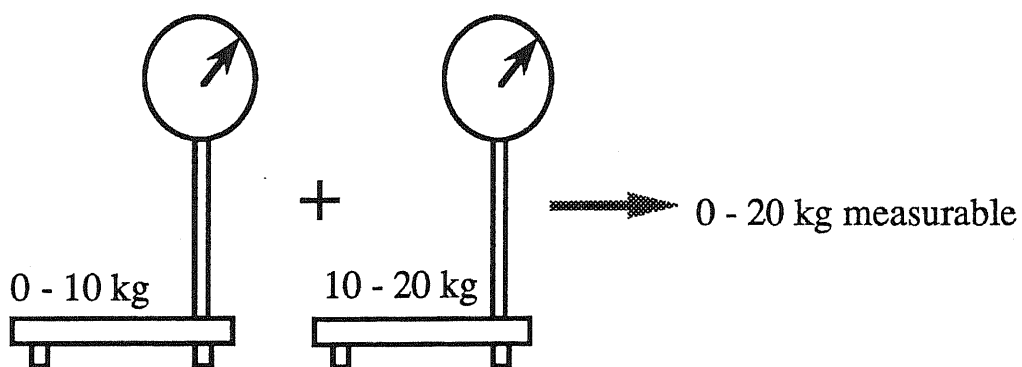
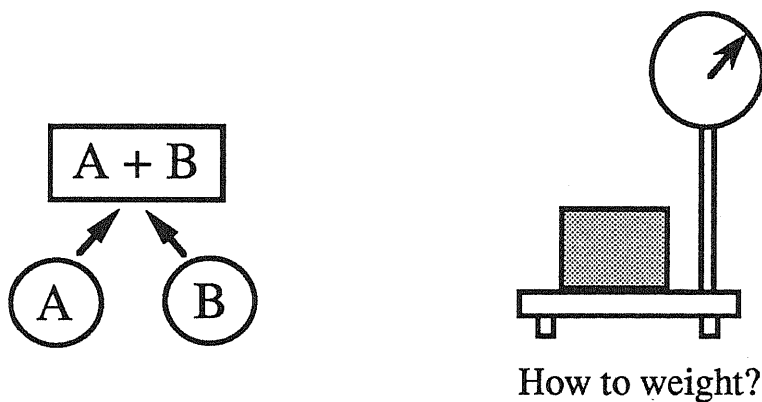
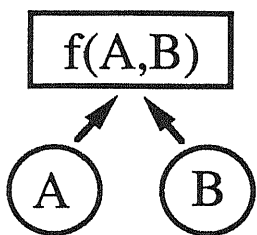


Figure 2.4 Sensor complex

# Sensor Integration

Operation: composing totally or partially

Effect: improvement in preciseness, confidence



How to weigh?

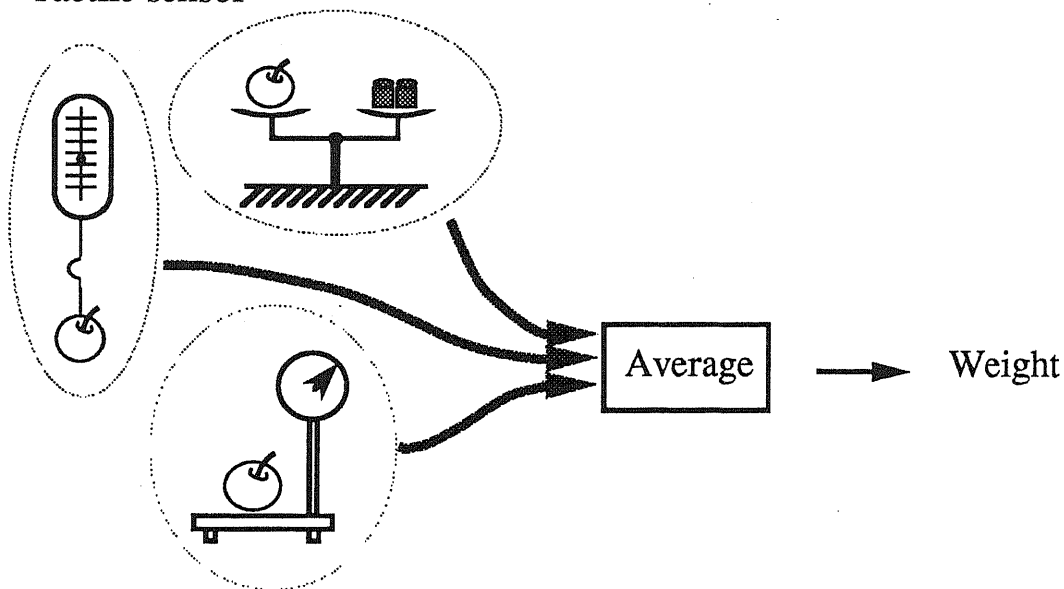
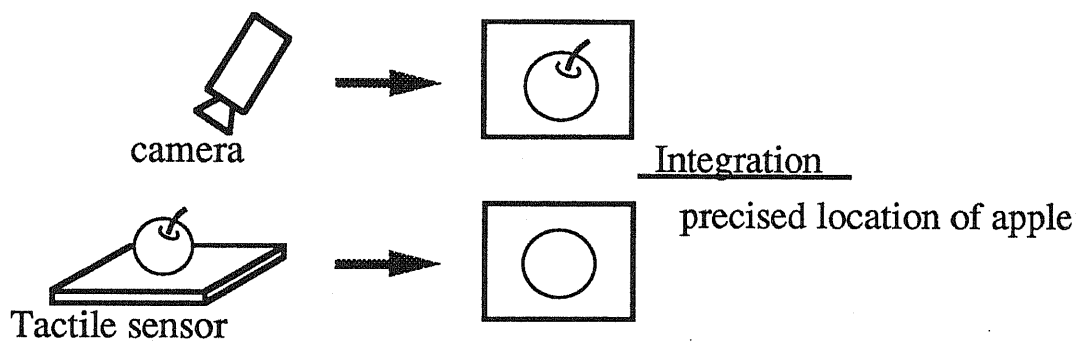


Figure 2.5 Sensor integration

Sensor Fusion

Operation: process mutually, cooperatively

Effect: obtain a new perceptual representation

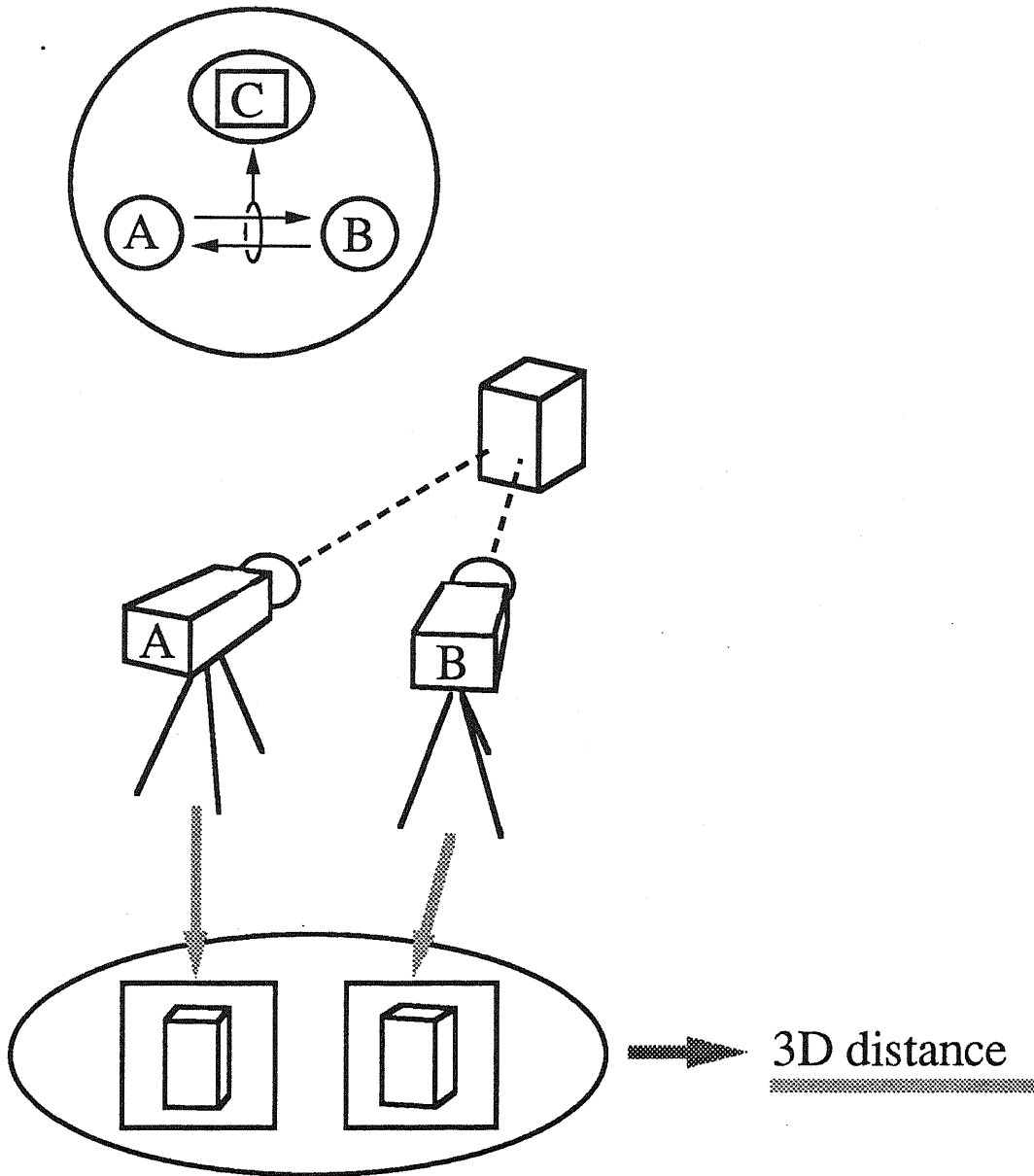


Figure 2.6 Sensor fusion

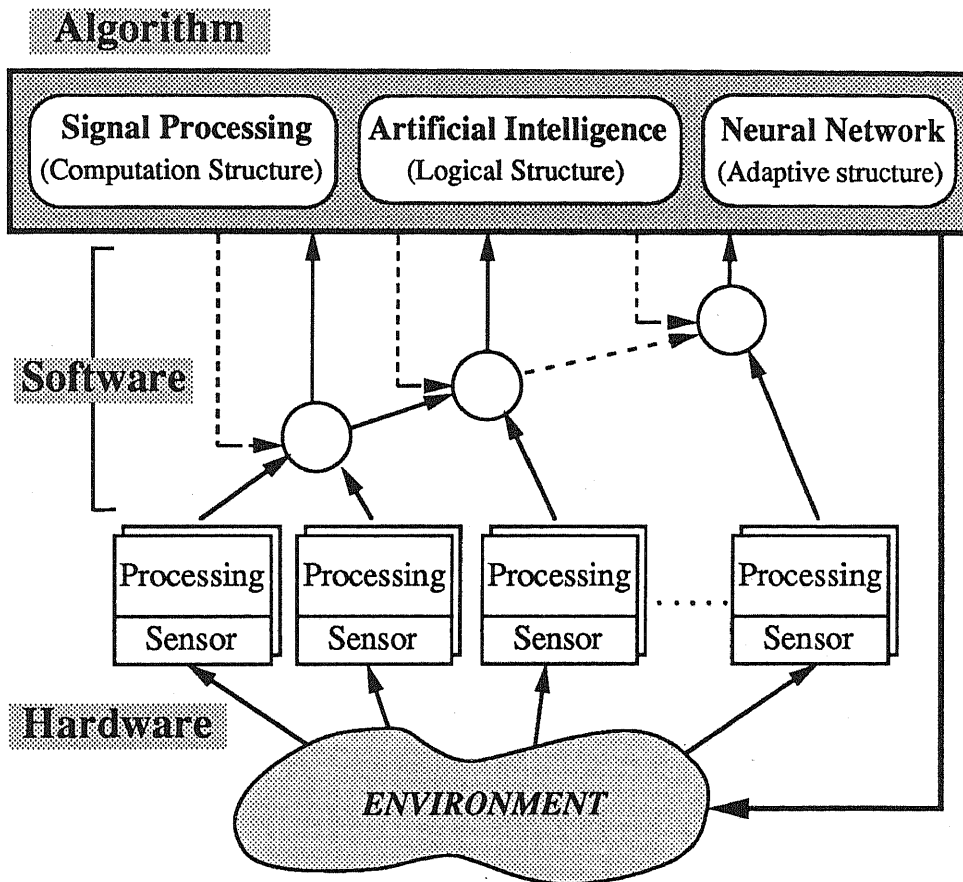


Figure 2.7 Structure of fusion system

## 参考文献

- [Ahmad86] S.Ahmad, "Real-Time Multi-Processor Based Robot Control," Proc. of IEEE Int. Conf. on Robotics and Automation, Vol.2, pp.858-863 (1986).
- N.Ayache, O.D.Faugeras, "Building, Registrating and Fusing Noisy Visual Maps," Proc. of 1st Int. Conf. of Computer Vision, pp.73-80 (1987).
- [Bhatt87] R.Bhatt, D.Gaw, A.Meystel, "A Real Guidance System for an Autonomous Vehicle," Proc. of IEEE Int. Conf. on Robotics and Automation, Vol.3, pp.1785-1791 (1987).
- [Brooks83] R.A.Brooks, "Solving the Find-path Problem by Good Representation of Free Space," IEEE Trans. on System, Man and Cybernetics, Vol.SMC-13, No.3, pp.190-197 (1983).
- [Crowly85] J.L.Crowly, "Navigation for an Intelligent Mobile Robot," IEEE Journal of Robotics and Automation, Vol. RA-1, No.1 (1985).
- [Dubourque86] V.Dubourque, H.Guiot, O.Ishacian, "Towards Multi-Processor and Multi-Robot Controllers," Proc. of IEEE Int. Conf. on Robotics and Automation, Vol.2, pp.864-870 (1986).
- [池上 89] 池上, 加藤, 大園, "自律移動ロボットのための境界距離モデルに基づく知識およびセンサ情報の表現," 電子情報通信学会論文誌D-II, Vol.J72-D-II, No.11, pp.1788-1799 (1989).
- [石川 90a] 石川, "センサフュージョンの現状と課題," 第29回計測自動制御学会学術講演会, pp.1-8 (1990).
- [石川 90b] 石川, "センサフュージョンの課題," 日本ロボット学会誌, Vol.8, No.6, pp.95-102 (1990).
- [井上 90] 井上, "ロボットの知能とシステム," 人工知能学会誌, Vol.1, No.1, pp.712-719 (1986).
- [柿倉 87] 柿倉, "自律移動のロボットにおける知能化技術(1) 行動プランニング," 日本ロボット学会誌5巻5号, pp.66-70 (1987).
- [柿倉 90] 柿倉, "自律型移動ロボットの経路探索問題," 電子情報通信学会誌, Vol.73, No.7, pp.744-753 (1990).

- [Khatib86] O.Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," *Int. Journal of Robotics Research*, Vol.5, No.1, pp.90-98 (1986).
- [木下 90] 木下, "センサ情報の統融合化に関する研究の現状," *日本ロボット学会誌*, Vol.8, No.6, pp.88-94 (1990).
- [小林 87] 小林, "自律型移動ロボットにおける知識処理," *計測と制御*, Vol.26, No.2, pp.128-133 (1987).
- [Kriegman87] D.J.Kriegman, E.Triendl, T.O.Binford, "A Mobile Robot : Sensing, Planning, and Locomotion," *Proc. of IEEE Int. Conf. on Robotics and Automation*, Vol.1, pp.402-408 (1987).
- [Leahy86] M.B.Leahy, G.N.Saridis, "The Ral Hierarchical Control System," *Proc. of IEEE Int. Conf. on Robotics and Automation*, Vol.1, pp.407-410 (1986).
- [Luo86] R.C.Luo, R.S.Scherp, M.Lanzo, "Automated Decision Tree Generation for Object Recognition and Classification," *Proc. of IEEE Industrial Electronics Society Conf.* pp.357-362 (1986).
- [Luo89] R.C.Luo, M.G.Kay, "Multisensor Integration and Fusion in Intelligent Systems," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol.19, No.5 (1989).
- [増田 87] 増田, "ロボットのセンシングシステム," *電気学会論文誌*, Vol.107-C, No.5, pp.430-435 (1987).
- [宮崎 85] 宮崎, 有本, 竹垣, 前田, "人工ポテンシャルに基づいたロボットマニピュレータのセンサフィードバック制御," *計測自動制御学会論文集*, Vol.21, No.1, pp.71-77 (1985).
- [Montgomery86] M.Montgomery, A.Meystel, "Navigation Algorithm for a Nested Hierarchical System of Robot Path Planning among Polyhedral Obstacles," *Proc. of IEEE Int. Conf. on Robotics and Automation*, Vol.3, pp.1616-1622 (1986).
- [Moravec83] H.P.Moravec, "The Stanford Cart and the CMU Rover," *Proc. of IEEE*, Vol.71, No.7, pp.872-884 (1983).
- [森 87] 森, 小谷, 石黒, 安富, 茅野, "構内道路を走行する自律移動ロボット," *日本ロボット学会誌*, Vol.5, No.5, pp.361-374 (1987).
- [Nakano85] Y.Nakano, et al., "A Concept for an Autonomous Mobile Robot and Prototype Model," *Proc. of ICAR*, pp.413-420 (1985).

- [Nilson69] N.J.Nilson, "A Mobile Automation : An Application of Artificial Intelligence Techniques," Proc. of 1st IJCAI, pp.509-520 (1969).
- [登尾 89] 登尾, 浪花, 有本, "クワッドツリーを利用した移動ロボットの高速経路生成アルゴリズム," 日本ロボット学会誌, Vol.7, No.5, pp.403-413 (1989).
- [奥富 83] 奥富, 森, "ポテンシャル場を用いたロボットの動作決定," 日本ロボット学会誌, Vol.1, No.3, pp.226-232 (1983).
- [Rehg] J.A.Rehg, "Introduction to Robotics," Prentice-Hall. Inc.
- [Rich] E.Rich, "Artificial Intelligence," Mcgraw-Hill Inc.
- [Saridis85] G.N.Saridis, "An Integrated Theory of Intelligent Machines by Expressing the Control Performance as Entropy," Control-Theory and Advanced Technology, Mita Press, Vol.1, No.2, pp.125-138 (1985).
- [佐藤 87a] 佐藤, 長谷川, "視覚フィードバックによるロボットの制御," 計測と制御, Vol.26, No.2 pp.122-127 (1987).
- [佐藤 87b] 佐藤, "ラプラスポテンシャルを用いた多次元空間内の障害物回避," 第5回日本ロボット学会学術講演会予稿集, pp.155-156 (1987).
- [坪内 87] 坪内, 油田, "地図とカラー画像を利用した屋内環境を移動するロボットのための視覚システム構成法," 電子情報通信学会論文誌D, Vol.J70-D, No.8, pp.1685-1688 (1987).
- [辻 86] 辻, "知識処理とロボティクス," 人工知能学会誌, Vol.1, No.1 (1986).
- [辻 87] 辻, "室内環境自律移動ロボットの視覚," 日本ロボット学会誌5巻6号, pp.42-46 (1987).
- [油田 87] 油田, "自律移動のロボットにおける知能化技術(2)ナビゲーションとガイダンス," 日本ロボット学会誌5巻5号, pp.71-74 (1987).
- [若松 a] 若松, 赤堀, 柿倉, "知能ロボット読本," 日刊工業新聞社.
- [若松 b] 若松, 田村, "ロボットの制御," 計測自動制御学会.
- [Wallance86] R.Wallance, et al., "Progress in Robot Road-Following," Proc. of IEEE Int. Conf. on Robotics and Automation, pp.1615-1621 (1986).



## Chapter 3

# 視覚情報を用いた 移動ロボットの経路探索

本章では、センサとして視覚センサを用い、階層的プランニングを行うことによって経路探索を行う移動ロボットコントロールシステムについて述べる。まず、移動ロボットシステムの構成について述べ、移動ロボットの重要な課題である経路探索を効率よく行うための階層的プランニングシステムを提案する。特に視覚センサを用いたローカルプランニング手法、回避ポイントの自動生成法、ナビゲーション法などについて詳しく述べる。

### 3.1 はじめに

移動ロボットにおいて、スタート地点から障害物を回避しながら目標地点に到達する経路を自動的に計画することは、ロボットがそなえるべき重要な機能である。経路探索の問題に関しては、多くの研究がなされており、さまざまなアルゴリズムが問題の状況に応じて提案されている [柿倉 86]。経路の探索方法は、環境についての知識がどれだけ事前に与えられているかに著しく依存する。環境に関する知識が完全な場合には、グラフ理論や数値計画法などを用いた方法が提案されている [柿倉 87]。一方、環境に関する知識が不完全な場合には、発見的知識を導入し、超音波レンジファインダを用いて探索する方法 [市川 84] や視覚センサを用いて位置の同定を行う方法 [Koch85] が提案されている。しかしながら、最適経路を求める論理的問題にとどまっていたり、リアルタイム性を失っているものが多い。最近、移動ロボットにおける各機能の使い方を考慮して、それらの機能を統合しようという研究 [Brooks85][Isik86] や高度な地理理解システムをもち、センサ情報によってリアルタイムに地図情報を更新する研究 [Wallance86][松本 86] も行われている。ここでは、環境に関する知識が不完全な場合の経路探索について考察する。環境に関する知識が部分的にしか与えられていない場合、知られている知識だけから大域的なプランニングを行い、未知の障害物に対しては実際に移動しながら局所的なプランニングを行うことによって対処する方法が考えられる。さまざまなセンサの中で、視覚は大域的情報を得る手段として盛んに研究されているが、距離情報の抽出に終始するものや、リアルタイム性を失っているものが多い。視覚情報の最大の特長は、情報の大域性にあり、それに着目して経路探索を行っている研究は少ない。

本章では、センサとして視覚センサを用い、階層的プランニングを行うことによって経路探索を行う移動ロボットコントロールシステムについて述べる。まず、移動ロボットシステムの構成について述べ、移動ロボットの重要な課題である経路探索を効率よく行うための階層的プランニングシステムを提案する [久保田 89b]。このシステムの中で、サブゴールマップ、単眼視による環境観測、視覚センサを用いたローカルプランニング、回避ポイントの自動生成法、ナビゲーション法などについて詳しく述べる。さらに3次元グラフィックスを用いたシミュレーションを行い、あまり複雑でない環境において本手法の有効性を示す。また、実験用移動ロボットを用いて、未知障害物回避実験を行い、提案したローカルパスプランニング手法の実験的検証を行う。

## 3.2 移動ロボットシステム

### 3.2.1 移動ロボットコントロールシステム

本研究では、移動ロボットコントロールシステムとして、Figure 3.1 に示すような4つのサブシステムからなる階層型システム構成を考える。

移動ロボットシステムにおいてプランニングシステムは、さらにグローバルプランナーとローカルプランナーからなる。グローバルプランナーはスタート地点からゴールまでの大域的なプランニングを行い、到達可能なサブゴールのシーケンスを出力する。ローカルプランナーはサブゴール間の局所的なプランニングを行う。ここでは、未知の障害物を回避するために視覚センサを用いて環境情報を取り入れ、回避ポイントを自動生成する。ナビゲーションシステムにおいて、ナビゲータは内界センサ情報と外界センサ情報とを利用して、地図上の自己の位置と方向を同定しながら、移動ロボットをローカルプランナーから与えられた回避ポイントに誘導する。経路の途中で遭遇する障害物やデータベースにない状況にたいしては適切に対処する。コントロールシステムはナビゲータからの命令に基づいて実際の移動機構系を制御する部分である。知覚システムはセンサから得られた信号を処理して、タスク実行や意思決定に必要なセンサ情報を作り出す。この4つのサブシステムのほかに移動用知識データベースがある。これはロボット用の地図にあたる環境に関する知識や目標物、障害物に関する知識を有するデータベースと認識や回避など実際に判断したり、行動する際のルールを示したルールベースからなっている。

### 3.2.2 階層型移動ロボットシステムの動作

階層型移動ロボットシステムの動作を Figure 3.2 に示す。移動ロボットは移動前に与えられた大域的な地図を用いてサブゴールマップを生成する。これはオフラインで行われる。実際に移動を行う際には、グローバルパスプランニングの結果得られたサブゴールのシーケンスにしたがって、サブゴールがひとつずつエグゼキュータに送られる。エグゼキュータは、グローバルプランナーから与えられたサブゴールに到達するようにセンサ情報を使って、実際に移動を制御する部分である。エグゼキュータは、ローカルプランナー、ナビゲータ、コントロールシステム、センシングシステムの4つの部分からなっている。ローカルプランナーは、障害物がある場合にはセンサからの情報にし

たがって、回避ポイントを生成し、その回避ポイントの情報をナビゲーションシステムに送る。そして、移動ロボットは、ナビゲータによって計算された経路に沿ってコントロールされる。移動ロボットが与えられたサブゴールに到着すると、エグゼキュータは到着を知らせるメッセージをグローバルプランナーに送り、次のサブゴール情報を受け取る。この一連の動作を繰り返すことによって、サブゴールを一つずつ達成し、最終的な目標点に到達することができる。

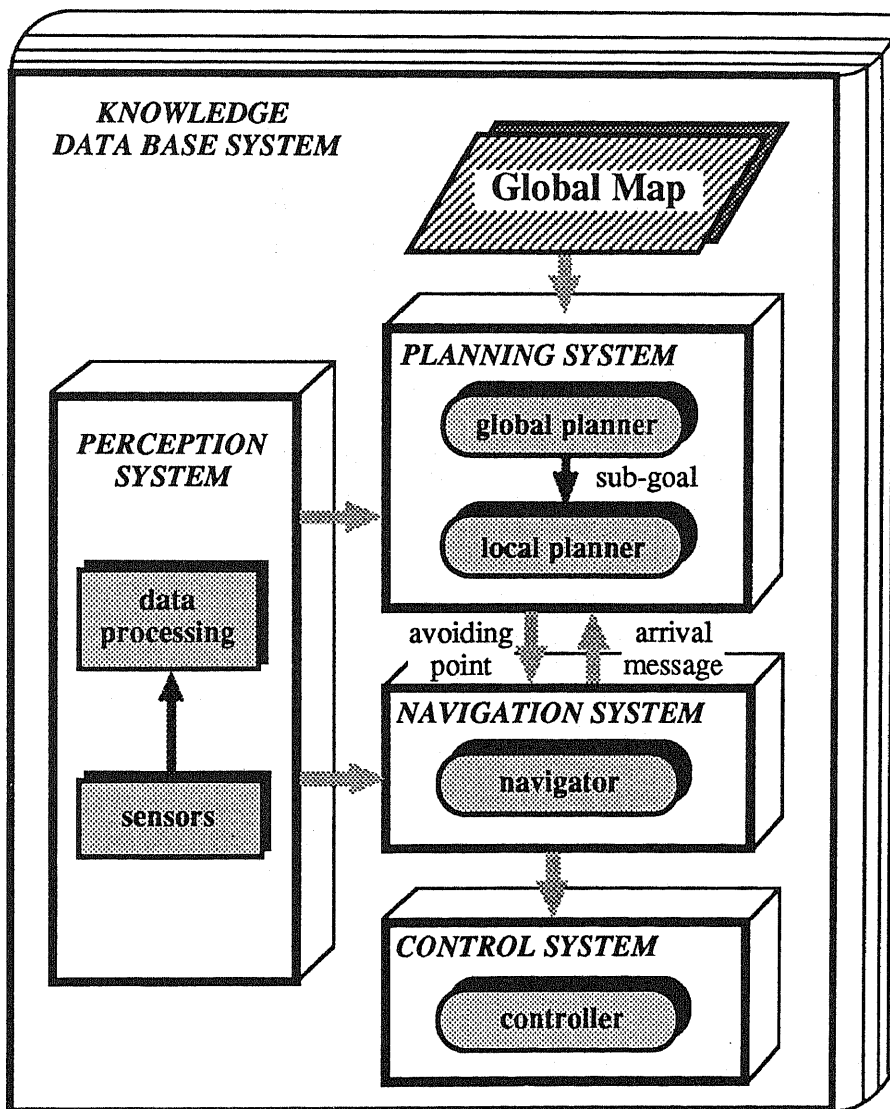


Figure 3.1 Mobile robot control system

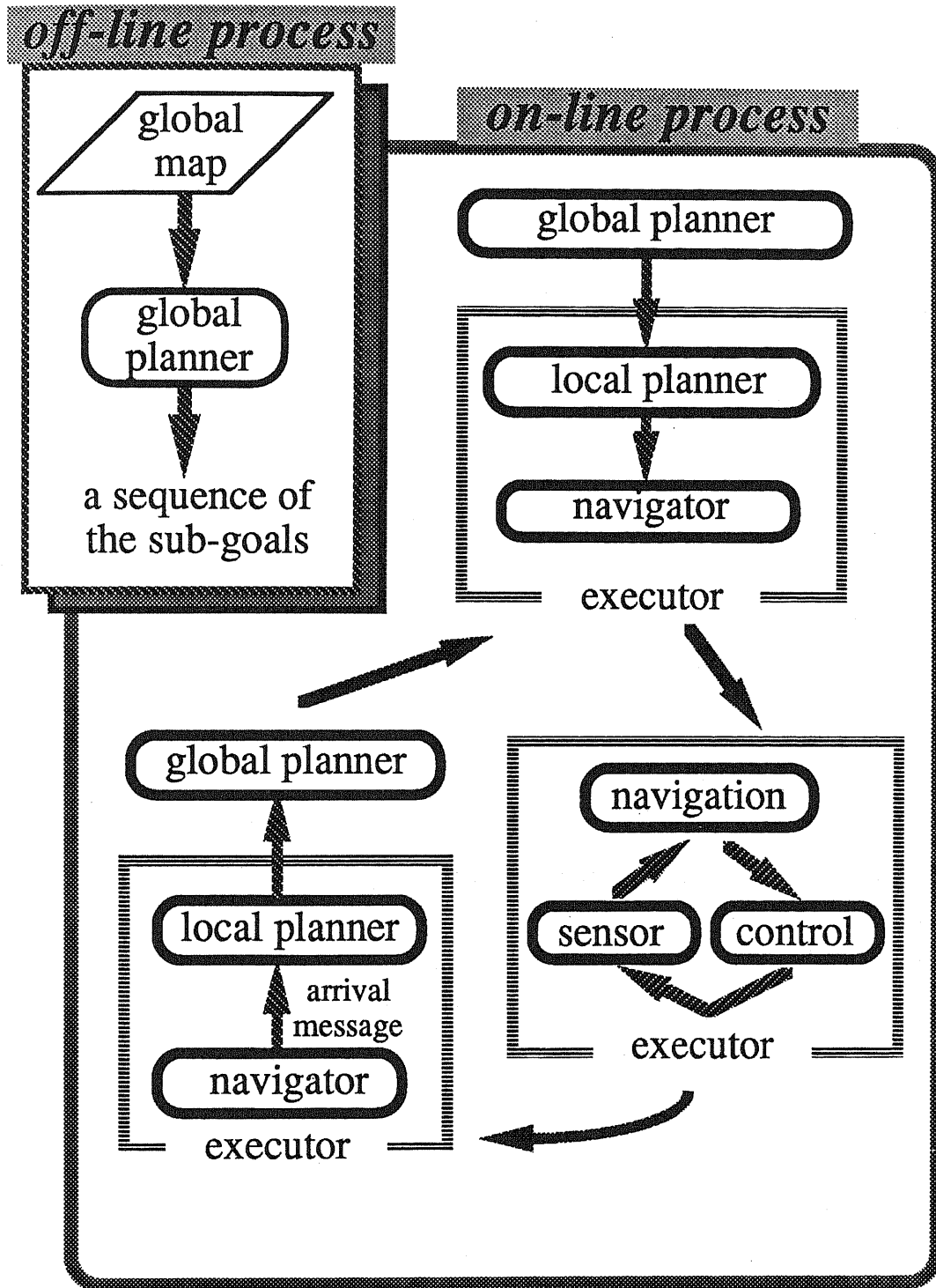


Figure 3.2 Operation of hierarchical system

### 3.3 階層型経路探索法

移動ロボットにおける経路探索問題とは、与えられた環境のもとで初期点から出発して、障害物を回避しながら目標点に到達するような経路を見いだすことである。移動ロボットの経路の探索方法は、環境に関する知識が事前にどれだけ与えられているかに大きく依存する。

本研究では、ロボットの移動する環境を屋内として考える。したがって、移動ロボットが働く世界は、ほぼ水平な床と垂直な壁にかこまれている。その中に置かれているものも、机や本棚といった人工物が多い。また、あらかじめ環境の構造や寸法などある程度知ることができる。さらに、自然環境に比べてイメージングの条件がよいなどの特徴も持っている。ここでは、環境に関する知識が不完全な場合を想定し、効率的な経路探索法として、次の階層的プランニング方式を提案する [久保田 87b]。

#### 3.3.1 階層的プランニング

ゴールまでの経路プランニングは Figure 3.3 に示すように次のような2つに分けることができる。1つはスタート地点からゴールまでどのような経路を選んだらよいかというような移動前に行えるグローバルパスプランニングと、もう1つは障害物回避などの実際に移動しながらロボットの環境に応じて行うローカルパスプランニングである。環境全体の地図を利用しなければならないのはグローバルパスプランニングにおいてであり、センサ情報を利用するのはローカルパスプランニングにおいてである。従ってこの2つのプランニングを階層的に行うことによって効率的なプランニング、及び環境地図、センサ情報の利用が可能になると考えられる。

#### 3.3.2 グローバルプランナー

移動ロボットは実際の移動を行う前にグローバルパスプランニングを行う。ここで、グローバルプランナーは人間から与えられた大域的な地図を利用する。大域的な地図とは移動ロボットが行動する領域をあらかじめ、教示して記述したもので、壁や机などの床面への射影である。

移動ロボットは移動環境全体に関するグローバルな知識や情報に基づいてサブゴー

ルマップを作成する [Harashima87]。サブゴールマップは、視覚的に確認し、追従できる目標（サブゴール）の集合からなる。各サブゴールには、そのサブゴールの視覚的特徴、及び、隣接するサブゴールの方向や距離などの情報をもっている。グローバルプランナーでは、最短距離や最短時間あるいは行きやすさなどのパラメータを考慮して、それらを合わせた評価関数を用いてスタート地点からゴールに至る大域的なプランニングを行う。サブゴールマップの概念をネットワークで表したものが、Figure 3.4 に示されている。Figure 3.4 において、ノードがサブゴールを表し、リンクでつながれているサブゴールはお互いに隣接していることを表している。グローバルパスプランニングは、スタート地点のノードとゴール地点のノードを結ぶ一連のリンクをある評価関数にしたがって決めることに対応する。サブゴールとしては、視覚的に確認できるものならどんなものでも使える。例えば、ドアや窓のセンターを画像で検出して用いることができる。あるいは、画像処理に負担のかからないものとして特殊なランドマークを用いることもできる [Tachi84]。

### 3.3.3 ローカルプランナー

ローカルプランナーは、グローバルプランナーから得られるサブゴール情報を基にして、実際にサブゴールとサブゴールを結ぶパスを設計する。視覚センサによってサブゴールが検出できる場合には、その目標に直進するようにコントロールしてやればよい。まず最初に画像を視覚センサによって取り込み、目標の画像面上の位置を検出する。カメラが移動ロボットの進行方向に固定されているとすると、画像面の中央の鉛直方向の直線が、移動ロボットの進行方向に対応する。したがって、目標の中心点がこの鉛直方向の直線に一致するように移動ロボットの移動をコントロールしてやればよい。このコントロールはおもに移動ロボットのステアリング調整によって行われることになる。

ローカルパスプランニングがもっとも大事になってくるのは経路上に、グローバルパスプランニングの際には考慮に入っていなかった障害物などがあって、与えられたサブゴールが確認できない場合である。この場合には、ローカルプランナーは移動しながら、視覚センサを用いて到達可能な障害物回避ポイントを生成する。移動ロボットは移動しながら未知の障害物に会うたびごとにローカルプランナーにより回避ポイントをもうけ、移動を繰り返す。サブゴールに到達したならば、到達メッセージをグローバルプランナーに送り、次のサブゴールの情報をもらう。以上のことを繰り返すことにより、最終的なゴールに到達することができる。

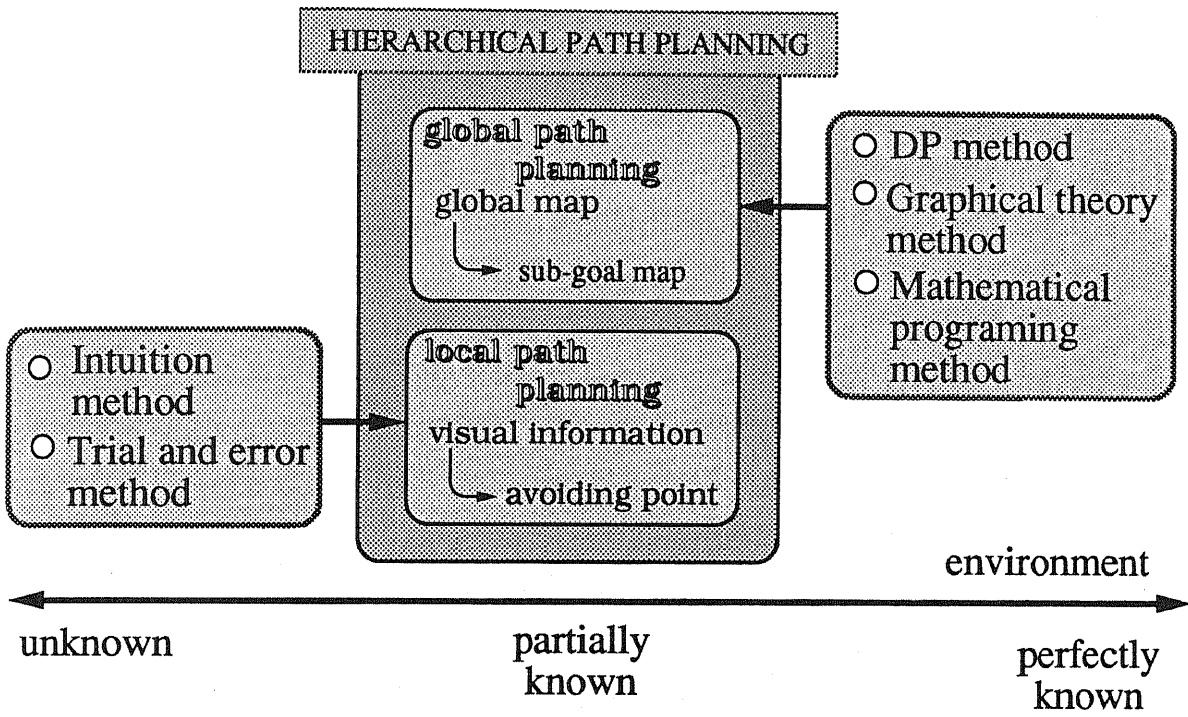


Figure 3.3 Hierarchical path planning

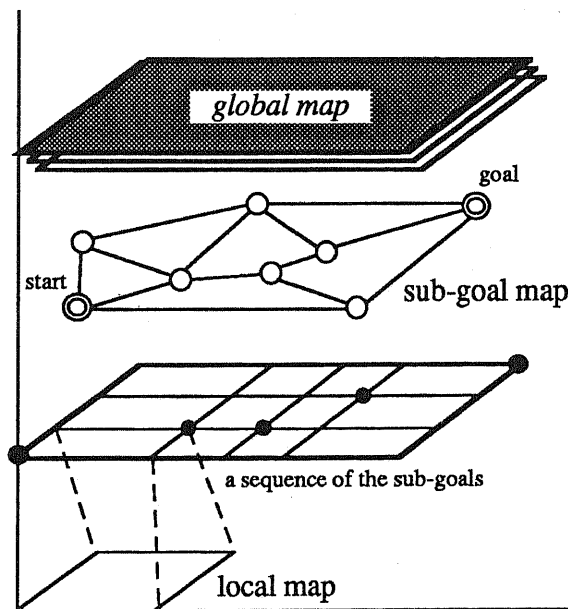


Figure 3.4 Hierarchical map



## 3.4 グローバルパスプランニング

グローバルパスプランニングとは、第(3.3.2)節で述べたように、人間から与えられた大域的な地図情報をもとにスタートとゴールとのパスをサブゴールのシーケンスという形で生成することである。これは人間が目的地に向かうとき、過去の記憶あるいは地図から目的地までの途中にある認識可能な目印を思い浮かべることに対応する。人間がロボットに与える環境地図情報としては、いろいろなデータ構造が考えられる。すでに目標となるドアや柱などがランドマークとして与えられていればそれをサブゴールとすればよい。また環境内の物体の形状や寸法がわかっているなど詳細な地図が与えられている場合には、物体の各頂点をサブゴールの候補とすることができる。ここでは、大域的な地図として天井に取り付けられたカメラで作業空間を撮影し得られた、本棚や机などの床面への射影である地図画像を用いる。これによりロボットは物体の位置や形状をオペレータの助けを借りることなく認識できる。グローバルパスプランニングは次の3つのステージからなる。(Figure 3.5 参照)

1. 大域的な地図情報（地図画像）が与えられる。
2. 地図画像から到達可能なサブゴールを生成する。
3. ゴールまでのパス（サブゴールのシーケンス）を決定する。

### 3.4.1 地図画像の利用

ここでは、大域的な地図として作業環境内の物体の床面への射影である地図画像が与えられているものとする。一般に画像の情報量は膨大なのでそれを直接探索しても高速に経路を生成できない。このような問題に対して、4分木を用いて2次元空間の領域を位置に関して階層的に分割・管理するクワッドツリーモデルが提案されている[Nilson69][登尾89]。ここでは、サブゴール生成のためにクワッドツリーを用いる。クワッドツリーでは、Figure 3.6に示すように障害物領域は黒ノード、自由領域は白ノード、それらが混在する領域はミックスノードで表現される。ミックスノードに対応する領域はさらに4つのサブ領域に分割され、ミックスノードの子ノードとして表現される。ロボットの大きさと安全余裕を考慮して、この分割は領域の大きさが移動ロボットの大きさ+安全余裕と同等になるまで行う。

クワッドツリーを用いる利点は以下のとおりである。

1. 任意の形状の物体に対処できる。
2. 作業空間内の障害物の形状が複雑でも探索空間としてのグラフが小さくなりアルゴリズムが高速化される。

### 3.4.2 サブゴールマップの生成

クワッドツリーによって地図画像が分割されたら、各自由領域の重心がサブゴールの候補となる。すべてのサブゴールを障害物領域と交差しないかをチェックしながら線分によって結ぶ。この線分が探索するグラフのアーキに相当し、各サブゴールがノードとなる。次に、移動ロボットの初期位置 ( $S$ ) と目標位置 ( $G$ ) を新たなノードとしてグラフに追加すると同時に、それぞれがリンク可能なサブゴールにアーキを接続する。以上の手順によってサブゴールマップが完成され、ロボットが移動可能な経路はグラフ上のリンクによって表される。

このサブゴールマップ上でグラフを探索し経路計画を行う際、アーキの長さなどをコストとして探索することによってサブゴールを結ぶ経路のうち最短な解を求めることができる。求められた経路 (サブゴールのリスト) について、可視性を考慮したフィルタリングを行う。すなわち、あるサブゴールからさらに先のノードが見える場合 (直線で到達可能な場合) 途中のサブゴールをリストから除外することができる。このようなフィルタリング処理をスタートからゴールまでに至る各サブゴールについて繰り返し行くと冗長なサブゴールを削除した最適な経路を求めることができる。この手続きは、経路が最適になるようにサブゴールを同じリンク上で稼働させながらチューニングを行うことに相当する。サブゴールマップからゴール到達パス (サブゴールのリスト) を生成する方法は従来のグラフ理論などの探索方法を用いることができ、その1つの方法を次節で述べる。

### 3.4.3 動的計画法によるグローバルパスの一決定法

出発点、目標点、サブゴールよりなるノードとそれらを結ぶリンクとからなるグラフで移動可能経路の集合が与えられた場合には、一般に動的計画法 (DP) を用いて経路を求めることができる [Barr81]。

いま、スタートを表すノードを  $n_1$ 、ゴールを表すノードを  $n_m$  とし、途中のサブゴールを表すノードを  $n_i$  とし、サブゴールの集合  $\{n_i\}(i = 2, 3, \dots, m-1)$  の中から経路を構成するノードシーケンスを求めるものとする。このとき、移動経路の評価関数  $F(n_1, n_2, \dots, n_m)$  は、

$$F(n_1, n_2, \dots, n_m) = f_1(n_1, n_2) + f_2(n_2, n_3) + \dots + f_{m-1}(n_{m-1}, n_m) \quad (3.1)$$

と表すことができる。ここで、 $f_i(n_i, n_j)$  は、2つのノードを結ぶ経路のコストである。

いま、スタートから  $n_i$  にいたる経路のコストを  $g_i(n_i)$  とすると、最適性の原理から

$$\begin{cases} g_{i+1}(n_{i+1}) = \min\{g_i(n_i) + f_i(n_i, n_{i+1})\} \\ g_1(n_1) = 0 \end{cases} \quad (3.2)$$

なる関数方程式が得られる。

$f_i(n_i, n_j)(i = 1, 2, \dots, m-1, j = 2, 3, \dots, m)$  は前もって与えられているから、 $g_m(n_m)$  すなわち、スタートからゴールまで最適経路をたどった時のコストを求めるためには、前式の関数方程式を利用し、 $g_1(n_1) = 0$  から始めて、あらゆる可能性に対して  $g_2(\quad), g_3(\quad), \dots$  を求めて表の形に整理して記憶しておき、 $g_m(n_m)$  から逆に最適解を選んでいけばよい。ここで、問題になるのはノード数  $n_i$  の数が多くなると表が膨大になり、計算の時間もかかるので、経路に関する制約条件や知識を利用して解の探索領域を制限することが必要になる。Figure 3.7 にマップのデータ構造の一例を示す。

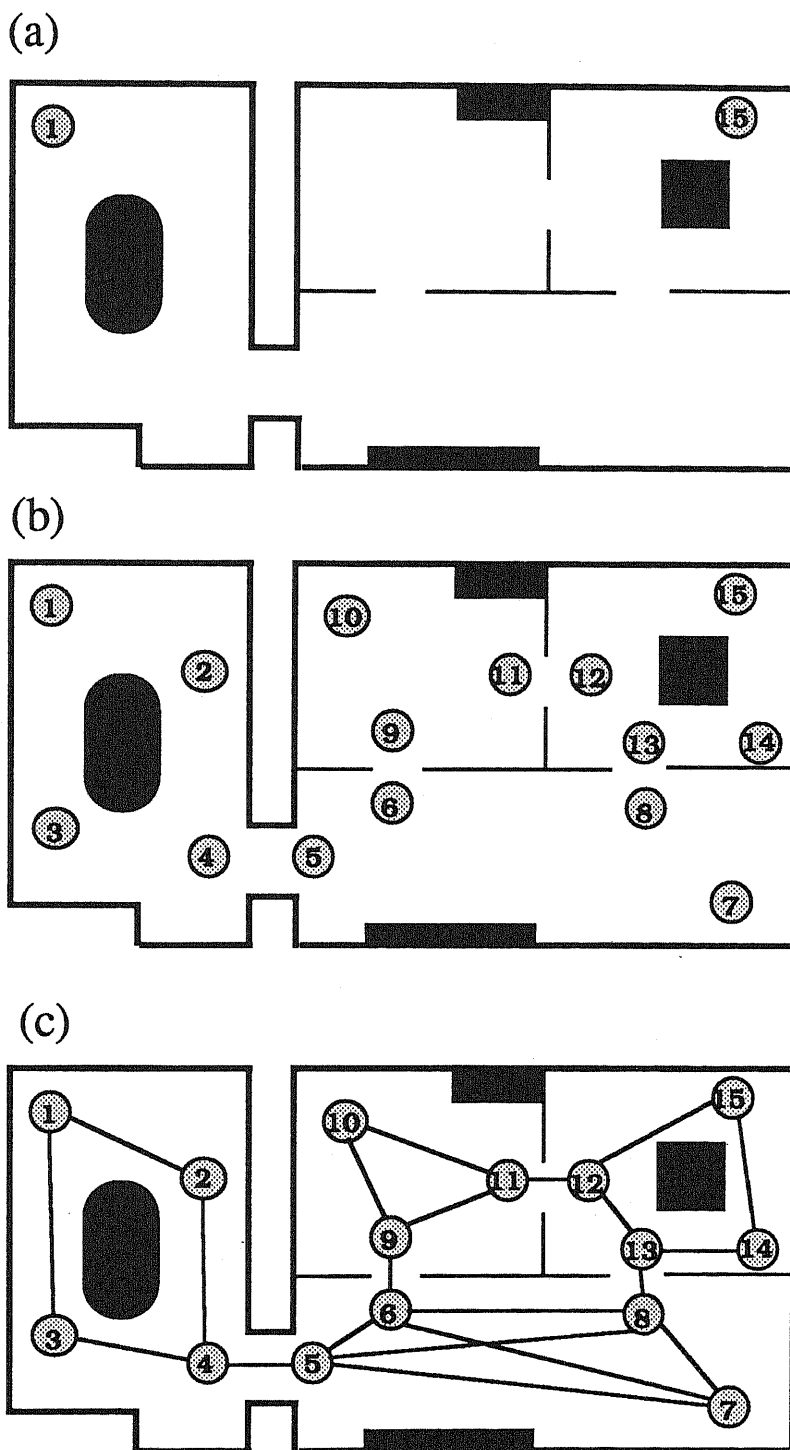


Figure 3.5 Image map data

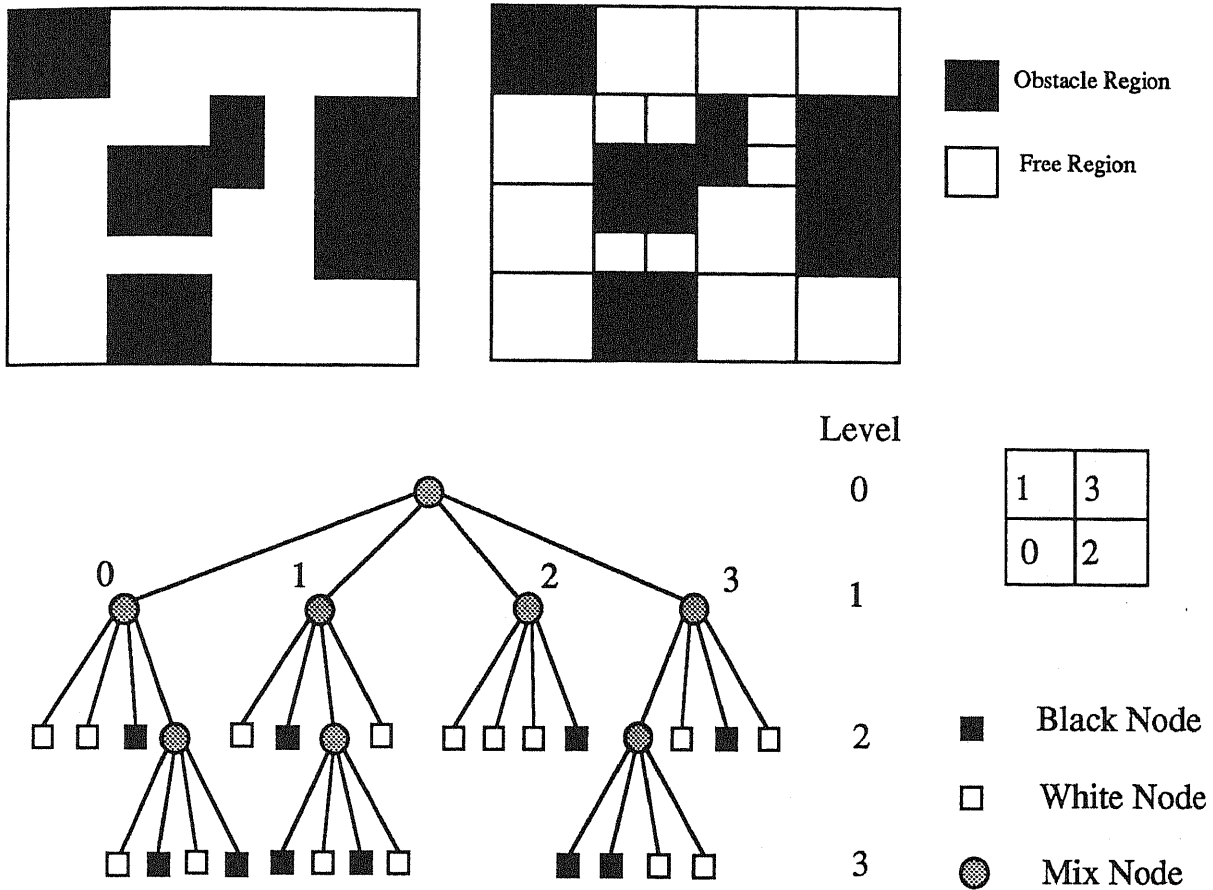


Figure 3.6 Quad tree method

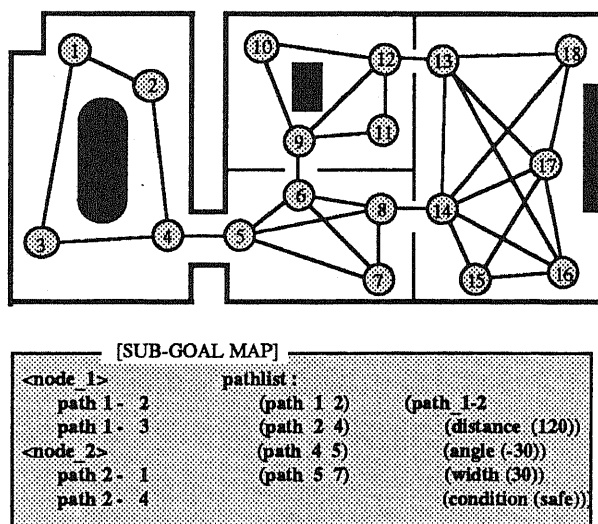


Figure 3.7 Data structure of sub-goal map

## 3.5 障害物回避動作

### 3.5.1 単眼視による環境観測

さまざまなセンサの中で、視覚は大域的情報を得る手段として、多くの研究がなされている。ロボットビジョンの研究[白井 a]において、視覚から環境の奥行き情報を抽出することは、中心的な課題となっている。しかし、視覚情報において、環境の奥行き情報は、基本的に欠落しており、画像から奥行き情報を得るためには、ステレオビジョン[Moravec83]、レンジファインダー[Shirai83]、オプティカルフロー[Tsuji85]などの特殊な手法を使う必要がある。また、影に着目して物体認識を行う方法[Raviv87]もある。しかし、これらの方法は左右画像の対応点の決定が困難であったり、各点の奥行き情報を抽出するため非常に長い処理時間がかかったり、リアルタイム性を失っているものが多い。また、それらの研究は、距離情報の抽出に終始するものが多い。視覚情報の最大の特長は、情報の大域性にあり、これを生かすような視覚センサの利用が望ましい。

本研究では、単眼視による環境観測を行い、視覚の与える大域的情報を利用して、移動ロボットの経路探索を考える。一般に、静止した1台のカメラの画像から、奥行き情報は得られない。そこで、「環境内の物体は床面上に立っていて、その最下部が見えている」という発見的知識を用いて、3次元世界での位置や大きさの算定を行う。例えば、物体が床面上にあると、カメラのパラメータと各点の画面内の位置から逆透視変換によって各点の位置を求めることができる。移動ロボットの進むべきゴールの位置は、あらかじめ与えられており、視覚によって認識できるものと仮定する。障害物に関しては、まわりの環境となんらかの色、明暗の変化がある（視覚的に検出可能である）ものとする。

対象（ロボット座標軸平面上）から画像面への変換は、焦点から焦点距離だけ離れた光軸上の点を含み、光軸に直交する平面への投影変換として表される[原島86]。Figure 3.8 は対象（床面上）と画像面との関係性を表したものである。したがって、移動ロボットの進行方向と一致する直線  $PP'$  は画面の鉛直方向のセンターライン  $pp'$  に対応し、進行方向と直交する直線  $QQ'$  は、画像面上ではある水平線  $qq'$  に対応する。そして、移動ロボットからの距離  $Y$  と距離  $Y$  にある線分の長さ  $D$  は幾何学的な関係（相似）を用いて次のような式で表される。

$$Y = \frac{H(K_1F - h \tan \beta_0)}{K_1F \tan \beta_0 + h} \quad (3.3)$$

$$D = \frac{K_2d(Y \cos \beta_0 + H \sin \beta_0)}{F} \quad (3.4)$$

ただし、

- $\beta_0$  : カメラの傾斜角
- $F$  : 焦点距離
- $H$  : カメラの床面からの高さ
- $d$  : 画像上での線分の長さ (画素数)
- $h$  : 画像上での対応する水平線の位置
- $K_1$  : *constant*
- $K_2$  : *constant*

ところで、単眼視による方法は簡単であるが、観測結果から作成した移動計画には問題がある。例えば物体が床から浮いていると、障害物の算定位置が実際の位置より遠くになり、衝突の危険性が生じてくる。また、床面上のマークや影なども障害物と見なして、それらを回避する計画を立てる可能性もある。したがって、これらを解決するために視覚センサのほかに他のセンサを併用して環境に関する正確な情報を獲得する必要が生じてくる [久保田 87a]。複数のセンサからの情報処理に関しては、第4章で詳しく述べる。

### 3.5.2 回避ポイントの決定方法

視覚情報から障害物の位置を検出し、それに応じて回避ポイントを決める [Kubota88b]。Figure 3.9(a) は移動ロボットが視覚センサから得た画像を示している。移動ロボットは画像からエッジポイントを検出する。ここで、エッジポイントとは画像の濃淡が近傍の点と大きく異なる画素をいう。まず、画像を水平方向にサーチし、左右のエッジポイントのうち最も左 (右) にあるポイントを通る垂直線を障害物の左 (右) エッジ (*left-edge (right-edge)*) として認識する。つぎに、画像を垂直方向に下からサーチし、画像の左半平面 (右半平面) において最も下にあるポイントを通る水平線を *left-distance-edge(right-distance-edge)* として認識する。この *left(right)-distance-edge* の画像上での位置を用いて、移動ロボットから障害物までの距離を計算する。回避ポイン

トは移動ロボットの幅と安全余裕を考慮して、画像上で  $s_L$  あるいは  $s_R$  におく。どちらにするかは画像のセンターラインからの変位  $a_L$ 、 $a_R$  の大小関係によって次のように決定される。

$$a_L \geq a_R \text{ のとき 回避ポイント} = s_R$$

$$a_L < a_R \text{ のとき 回避ポイント} = s_L$$

となる。

このように選ぶことにより、ステアリング角度がより小さくてすむ。ロボット座標軸上であらわすと Figure 3.9(b) のようになる。回避ポイントの位置を決める回避幅  $\Delta$  は次の式で表される。ここで、 $\eta$  の値は実験的に求めている。

$$\Delta = \Delta_{ob} + \frac{1}{2}W + \eta \quad (3.5)$$

ただし、

- $\Delta_{ob}$  : 回避する障害物の幅
- $W$  : 移動ロボットの幅
- $\eta$  : 安全余裕

回避ポイント  $AP_i$  からみたサブゴール  $SG$  までの距離を  $L_i$ 、方向を  $\theta_i$  とすると次の回避ポイント  $AP_{i+1}$  からみたサブゴールの位置  $(L_{i+1}, \theta_{i+1})$  は次式で表される。

$$L_{i+1} = \sqrt{\{L_i \cos \theta_i - l_i \cos(\alpha_i + \theta_i)\}^2 + \{L_i \sin \theta_i - l_i \sin(\alpha_i + \theta_i)\}^2} \quad (3.6)$$

$$\theta_{i+1} = \tan^{-1} \frac{L_i \cos \theta_i - l_i \cos(\alpha_i + \theta_i)}{L_i \sin \theta_i - l_i \sin(\alpha_i + \theta_i)} \quad (3.7)$$



### 3.5.3 複数の障害物に対する画像処理アルゴリズム

障害物が1個の場合には、第3.5.2節で述べた方法にしたがって回避ポイントを生成することができるが、2個以上の障害物を回避する場合には多少の変更を必要とする[久保田 88a]。一般的な画像データは、例えば Figure 3.10(a) のようなものである。この場合に最初の回避すべき障害物は物体  $C$  であることは明らかである。したがって、とりあえず他の障害物（この場合物体  $A$  と物体  $B$ ）を無視して物体  $C$  のエッジに関する情報  $d, d_l, d_r$  を得る方法を考える。この情報をもとにして、あとは第3.5.2節で述べた方法で回避ポイントを生成する。

さて、エッジ情報の抽出方法であるが、いま座標系を Figure 3.10(b) のようにとるとき  $x$  軸の間隔  $\delta$  おきに  $y$  軸方向を逆にトレースして、物体の下端までの距離  $h_i$  を調べる。そして、 $|h_i - h_{i-1}|$  があるしきい値  $h_{th}$  を越えたとき、そこに物体の左または右のエッジがあるものとする。（Figure 3.10(b) では I、III、V の位置に物体の左端が、II、IV、VI の位置に物体の右端があることがわかる。）ただし、画像スキャン間隔  $\delta$  およびしきい値  $h_{th}$  は実験的に求めている。そのうちで、 $h_i$  の値が最小となるもののみ（Figure 3.10(b) では III、IV）を残し、さらに近傍を同様にして調べ、あるしきい値を越えたときの  $x$  の値を左端、右端の  $x$  座標とする。また、 $d$  については全ての  $h_i$  のうち最小のものを  $d$  とすればよい。Figure 3.10(c) で示すように、物体  $A$ 、 $B$ 、 $C$  それぞれについて詳細に探索することによってエッジを検出している。各物体に対し候補となる回避ポイントを生成した結果を Figure 3.10(d) に示す。このアルゴリズムによるエッジ検出関数 `search` は C 言語によって記述されており、この関数を用いることによって、一番近くにある物体を認識し、回避ポイントの生成に必要なエッジ情報 ( $d, d_l, d_r$ ) を得ることができる。

ただし、本アルゴリズムにおいて物体同士が近い位置関係にあり、重なり合っている場合にはしきい値  $h_{th}$  によって2つの物体を識別することができなくなる可能性がある。しかしながらここでは障害物回避を目的としているので、近距離位置関係にある2物体をひとかたまりの物体と認識し回避ポイントを生成しても問題ないと思われる。

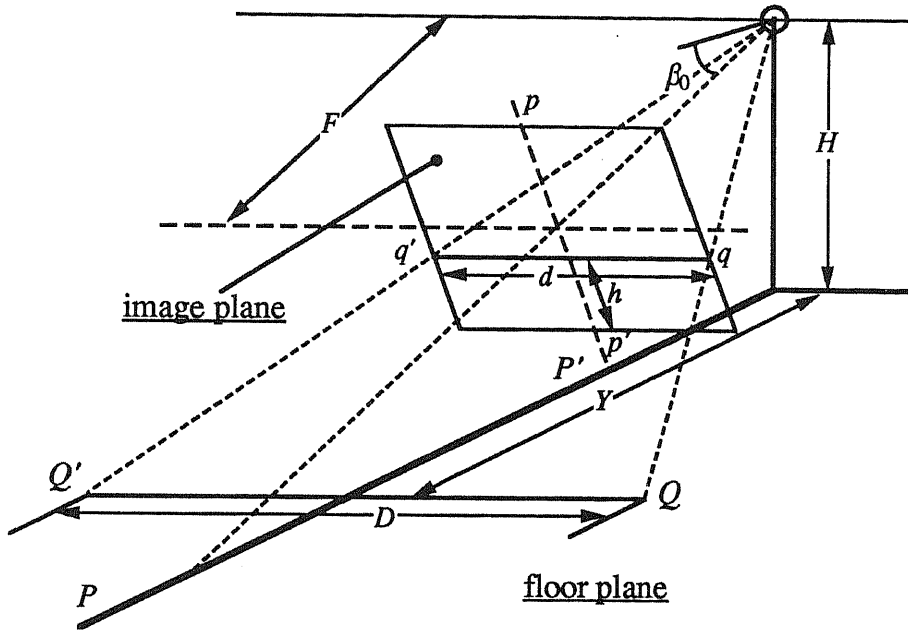


Figure 3.8 Relationship between floor plane and image plane

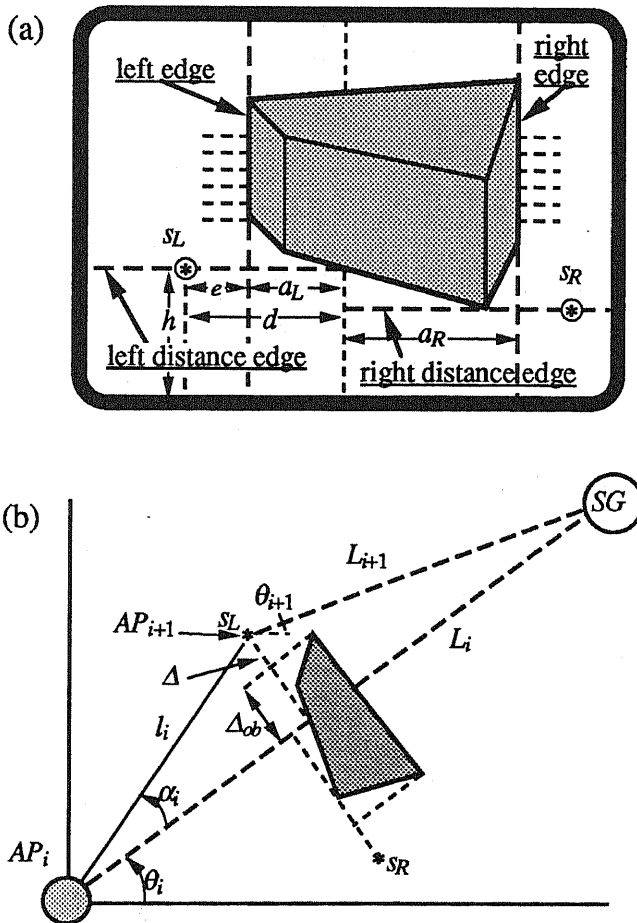


Figure 3.9 Method to decide avoiding point

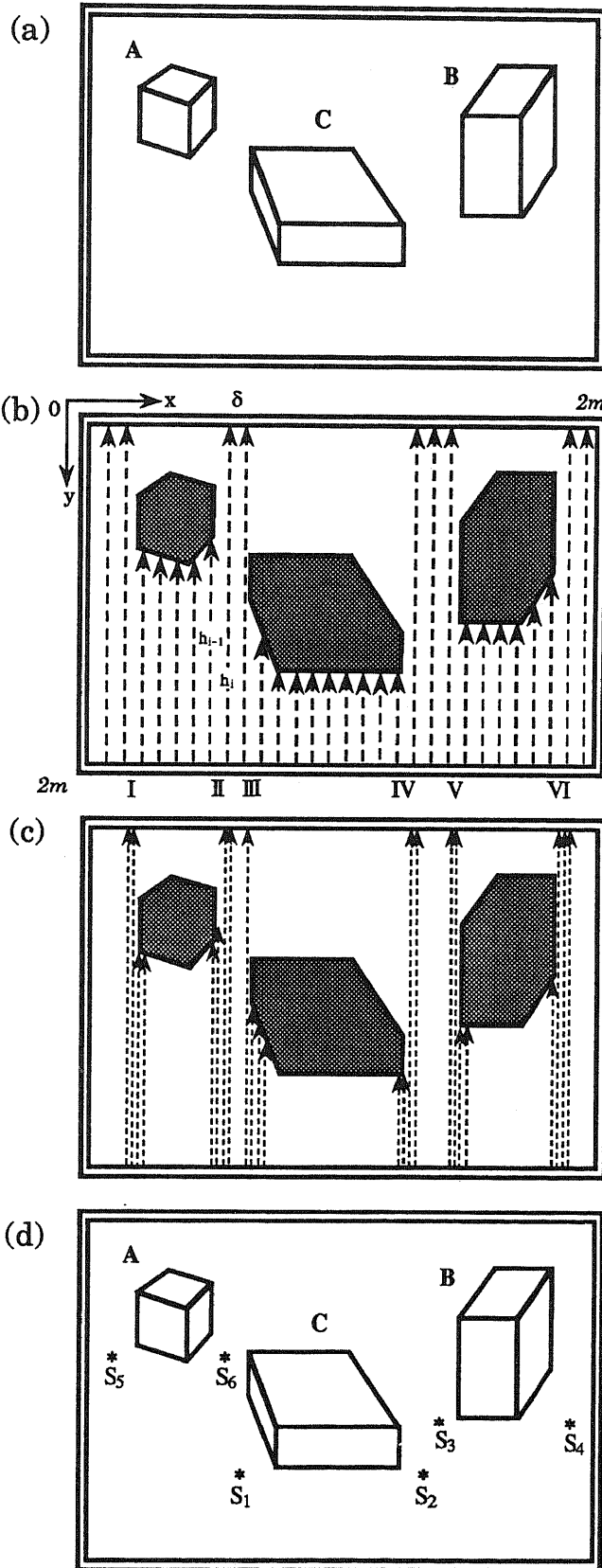


Figure 3.10 Image searching method

## 3.6 ローカルパスプランニング

### 3.6.1 アルゴリズム

ローカルパスプランニングアルゴリズム ( Figure 3.11 参照 ) を以下に示す。

#### <ステップ 0>

移動ロボットは、あるサブゴール  $ST$  から次のサブゴール  $SG$  に向かって移動するものとする。いま、 $i$  番目の回避ポイントを  $AP_i$  であらわす。また、回避ポイントの集合を  $S$  とする。初期設定として、 $i=0$  とし  $AP_0 = ST$  とする。また、 $S = \phi$  とする。

#### <ステップ 1>

移動ロボットは、サブゴールの方向を向く。視覚センサによってサブゴールが確認できるならば、<ステップ 4>にとぶ。障害物等のためサブゴールが確認できないならば、つぎの<ステップ 2>にすすむ。

#### <ステップ 2>

移動ロボットは、視覚センサから前述の方法によって次の回避ポイント  $AP_{i+1}$  を決定する。ただし、現在いる回避ポイント  $AP_i$  が回避ポイントの集合  $S$  に含まれている場合には前回と違う回避ポイントを選ぶものとする。

#### <ステップ 3>

移動ロボットは<ステップ 2>で生成された回避ポイント  $AP_{i+1}$  に移動する。つぎにもといた回避ポイント  $AP_i$  を回避ポイントの集合  $S$  の要素に入れる。 $i$  に 1 を加えて、<ステップ 1>にもどる

#### <ステップ 4>

サブゴールに向かって直進する。移動ロボットがサブゴールに到着したら、到着メッセージをグローバルプランナーに送る。

このアルゴリズムに関して、次の命題が得られる。

【命題 3.1】

(前提条件)

- [a] 移動ロボットはサブゴールの位置を既知とし、サブゴールに到達できるパスは少なくとも1つは存在するものとする。
- [b] 移動ロボットが動くエリアはサブゴールの位置を中心とし、 $ST$ と $SG$ とを結ぶ線を半径とする円内（ローカルマップ）とする。
- [c] 各障害物間はずべて移動ロボットの大きさに比べて十分大きいものとする。
- [d] ローカルマップ上には、有限個の障害物があるものとする。

このとき、移動ロボットはバックトラックすることなく、有限回、回避ポイントを生成することによってサブゴールに到達することができる。

【略証】

いま、Figure 3.12 に示すようにエリア  $C(i)$  をサブゴール  $SG$  を中心、回避ポイント  $AP_i$  とサブゴール  $SG$  を結ぶ線を半径とする円とし、そのなかに障害物が  $n_i$  個存在するものとする。ただし、エリア  $C(0)$  はローカルマップと一致する。

[1]  $n_i = 0$  のとき

<ステップ 1> と <ステップ 4> によってロボットはサブゴールに到達できる。

[2]  $n_i \neq 0$  のとき

<ステップ 2> によってロボットはつぎの回避ポイント  $AP_{i+1}$  をみつける。このとき障害物はおのおの十分離れているので、回避ポイントの方向  $\alpha$  に関して  $-\frac{1}{2}\pi < \alpha_i < \frac{1}{2}\pi$  となる。したがって、回避ポイントまでの距離  $L$  に関しては  $L_{i+1} < L_i$  となる。すなわち、移動ロボットは回避ポイントを生成するごとにサブ

ゴールに近づいていく。エリア  $C(i+1)$  の大きさはエリア  $C(i)$  の大きさよりも小さくなる。視覚センサからは障害物の奥行き情報が得られないので、回避ポイントを1回生成しただけではその障害物を完全に回避することができない。従って、最大  $m$  回 ( $m$ :有限) の操作を行うことにより、エリア内の障害物の個数を少なくとも1つ減らすことができる。すなわち、 $n_{i+m} < n_i$  となる。

いま、ローカルマップ ( $C(0)$ ) 上の障害物の個数を  $n$  とすると、[1]、[2] から  $\langle$ ステップ 2  $\rangle$  の操作を有限回 (たかだか  $mn + 1$  回) 繰り返すことにより、ロボットはサブゴールに到達することができる。

(注) このアルゴリズムによって生成されたパスは必ずしも最適なパスとは限らない。

### 3.6.2 仮想格子点平面における適用

ここでは、簡易な視覚情報処理で回避ポイントを生成するために、経路探索空間を床面の座標値を離散的に量子化した格子点平面に限定する [久保田 89a]。

回避ポイントの候補は、移動ロボットの幅と安全余裕を考慮して、床面の格子点に対応する画像上の仮想格子点とする。さらに、回避ポイントの候補と画像のセンターラインとの変位の大小関係によって優先順位をきめる。ここでは、障害物を回避し易いポイントを選ぶという理由から、センターラインに近い順に優先順位をつけ、最も近い点を第一の回避ポイントに選ぶ。

格子点平面における回避ポイント生成のための画像処理過程及び探索過程を Figure 3.13 及び Figure 3.14 にそれぞれ示す。本方式は、Figure 3.15 の探索木表現で示すように縦型経路探索を行っている。

格子点平面において、ローカルパスプランニングアルゴリズムを適用する場合、次の命題が得られる。

## 【命題 3.2】

(前提条件)

- [a] 移動ロボットはサブゴールの位置を既知とし、サブゴールに到達できるパスは少なくとも1つは存在するものとする。
- [b] 移動ロボットが動くエリアはサブゴールの位置を中心とし、 $ST$ と $SG$ とを結ぶ線を半径とする円内（ローカルマップ）とする。
- [c] ローカルマップ上には、有限個の障害物があるものとする。

このとき、移動ロボットは有限回、回避ポイントを生成することによってサブゴールに到達することができる。

## 【略証】

本アルゴリズムを各回避ポイントをノードとする2分木で表現する。ローカルマップの大きさ、障害物の個数、格子点平面という制約から、回避ポイントの候補総数は有限である。いま、ゴールに到達できる直前の回避ポイントは、少なくとも1つ以上存在するが、その中で一番深い節点の深さを  $d$  とすると、全節点数は最大  $2^{d+1} - 1$  である。従って、最大でも  $2^{d+1} - 1$  の節点を探索すればよい。さらに、<ステップ2>により、すでに通った回避ポイントを2度とおる場合には、不要な節点は切り落としているため、無限ループに落ちることはない。よって、有限回探索を行うことにより、必ずゴールに到達できる。

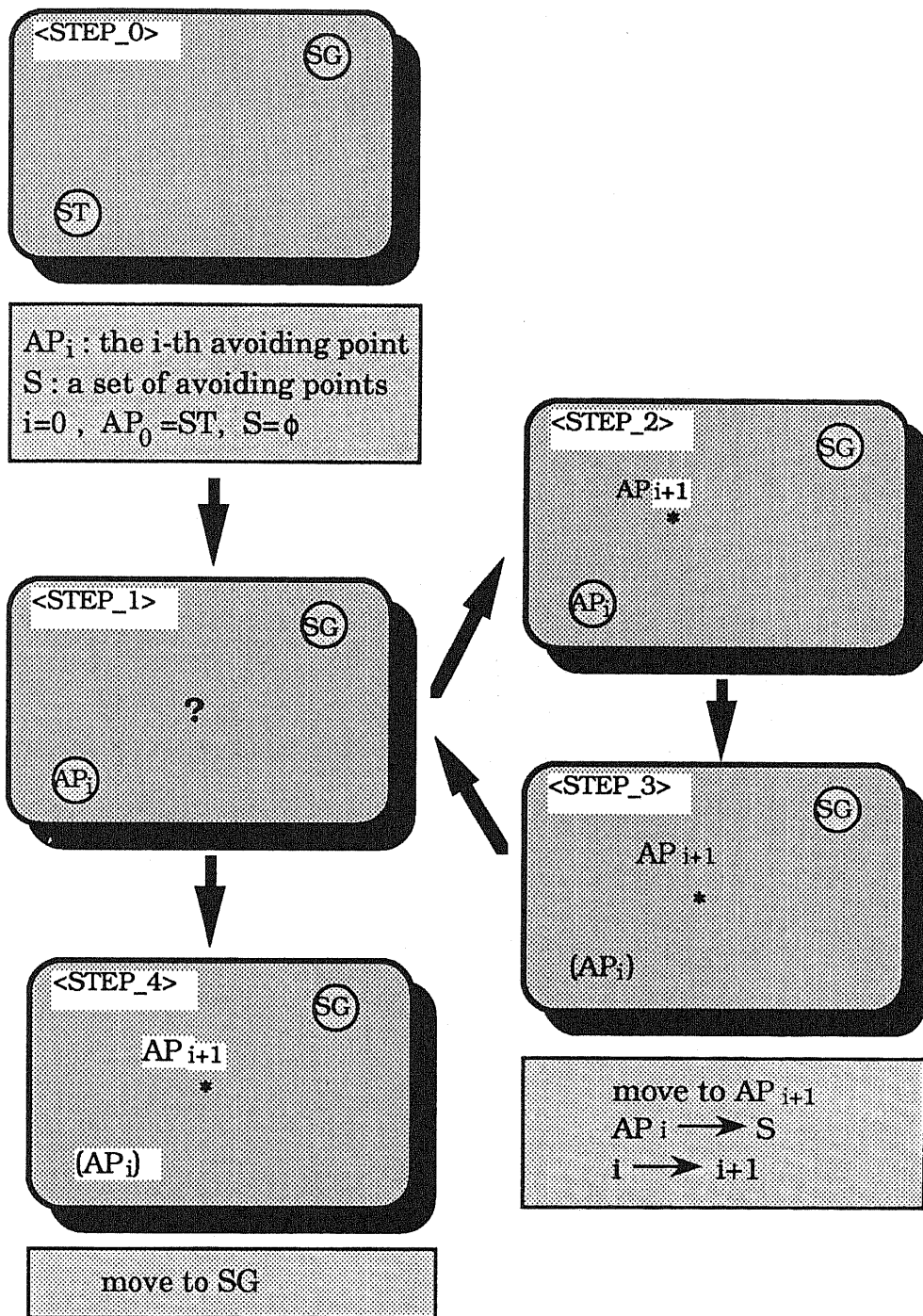


Figure 3.11 Local path planing algorithm



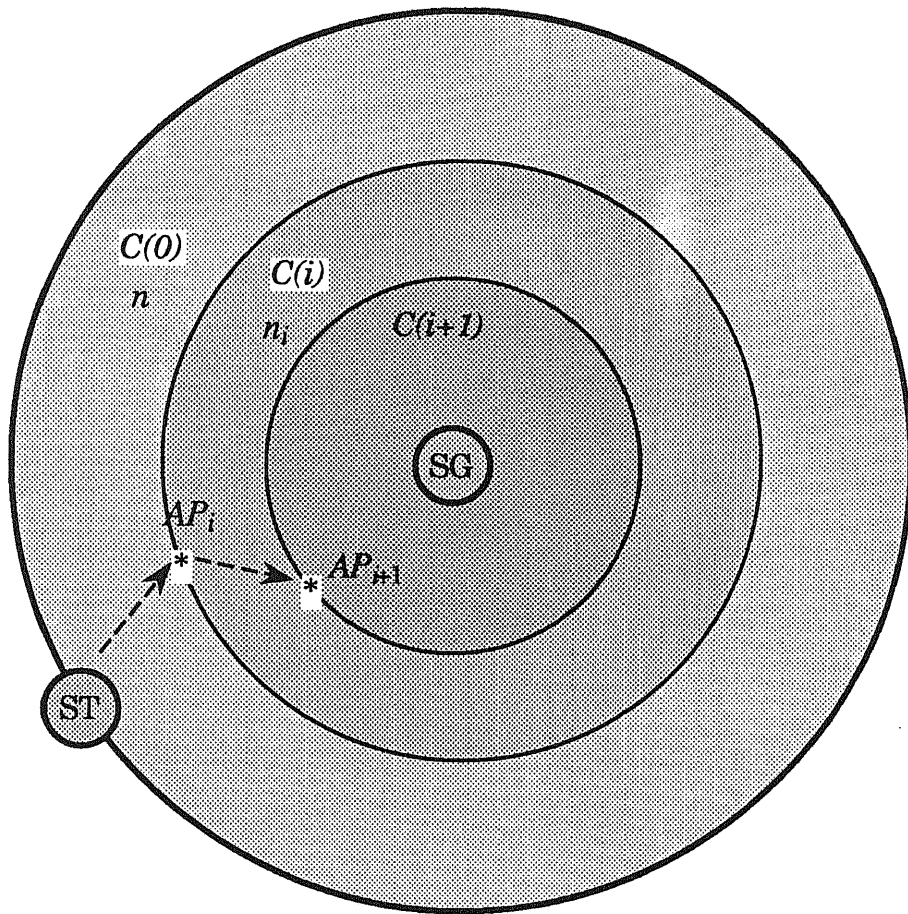
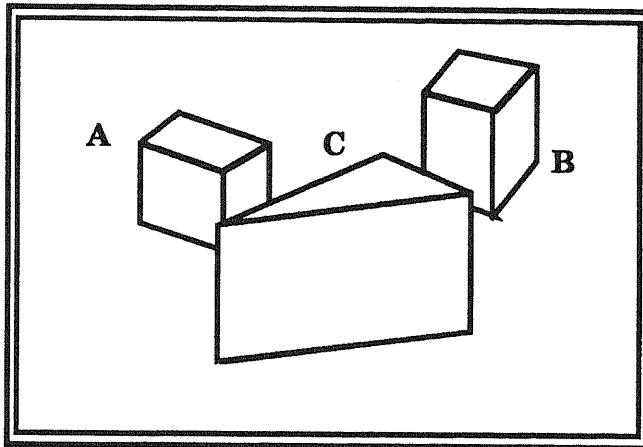
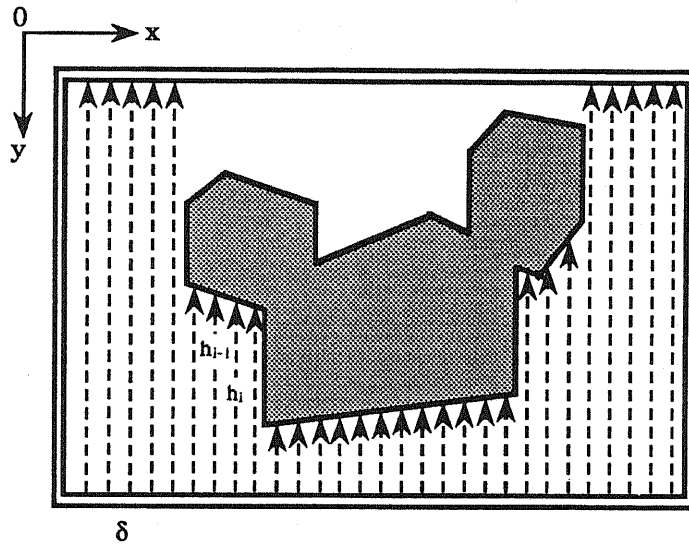


Figure 3.12 Local map

(a)



(b)



(c)

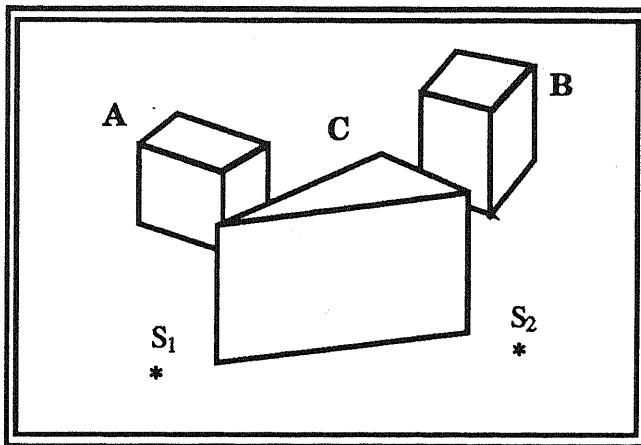


Figure 3.13 A method to generate avoiding points

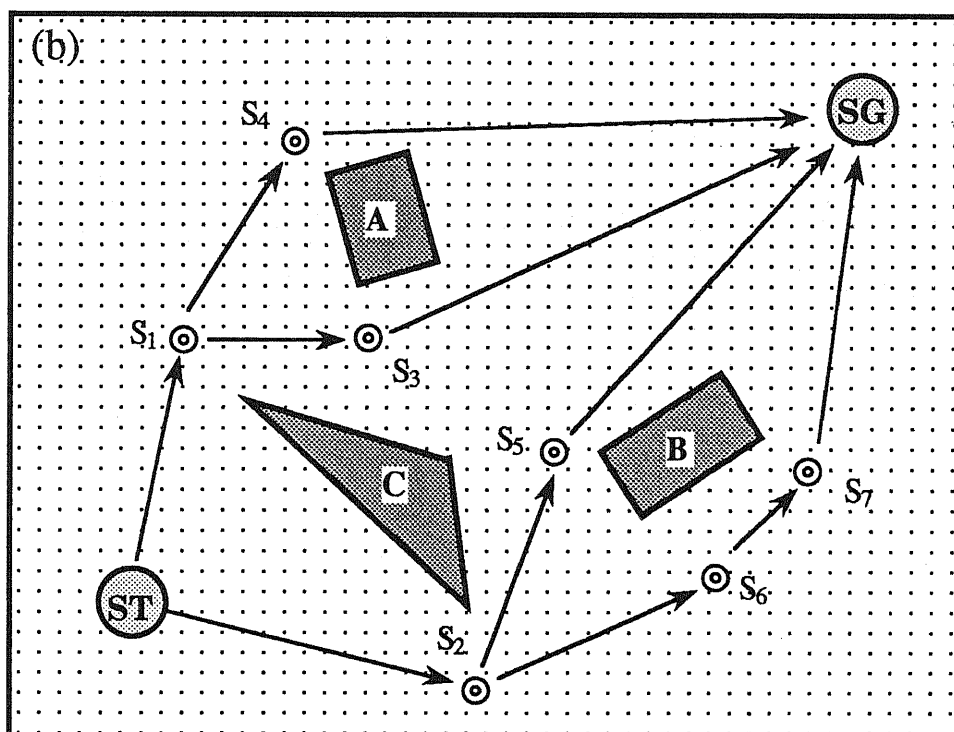
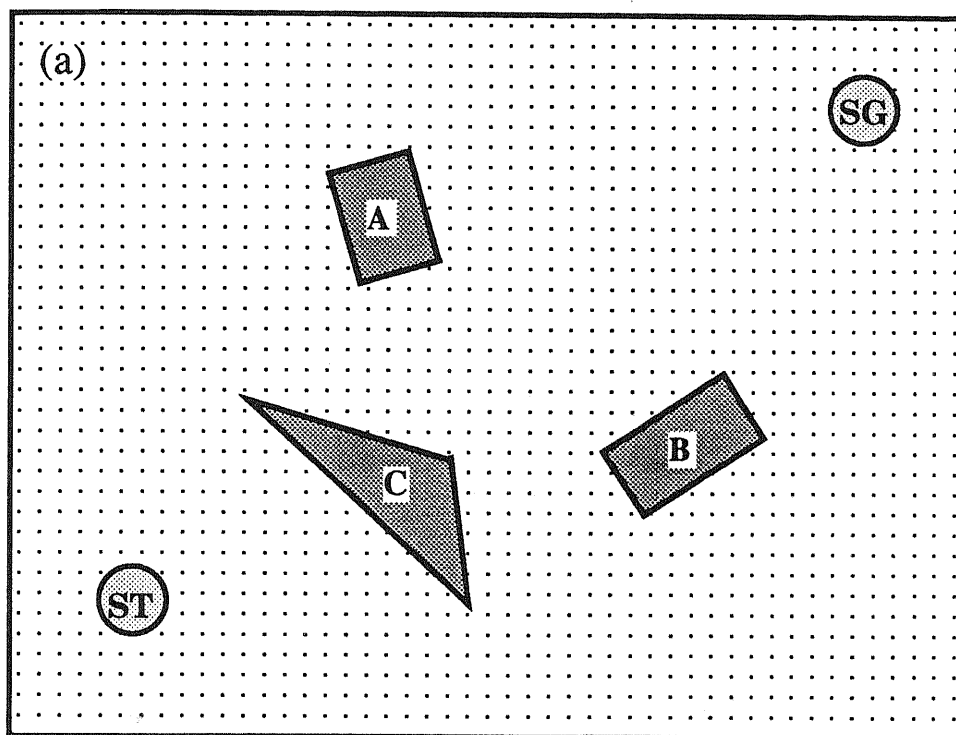


Figure 3.14 Vertical lattice point plane

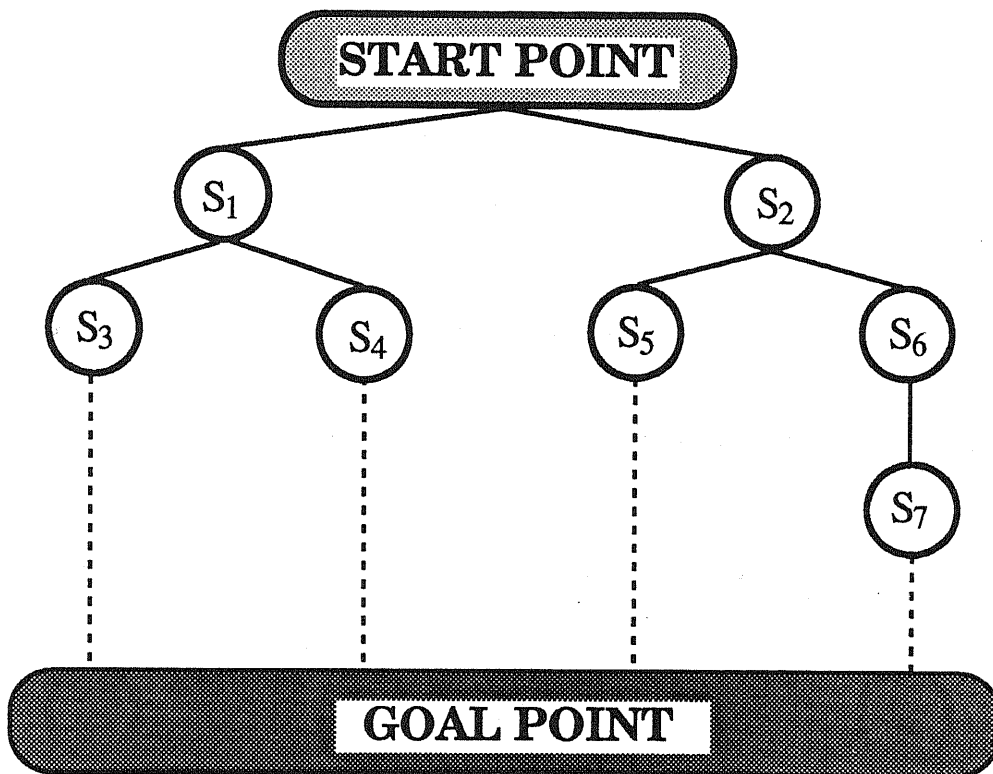


Figure 3.15 Searching tree

### 3.7 ナビゲーション

ナビゲータの役割は障害物に衝突することなく、ローカルプラナーによって生成された回避ポイントへ、移動ロボットを誘導することである。回避ポイントの位置情報はローカルプラナーから、ロボットの現在位置を中心に設定した2次元座標系内のベクトルで与えられる。また、これは視覚センサから得られた画像上の点としても与えられる。さらに、ローカルプラナーから与えられる情報はその回避ポイントの位置情報だけでなく、その点におけるサブゴールの位置情報も含まれる。したがって、到達すべき回避ポイントとその点において移動ロボットが持つべき速度（姿勢）を考慮して経路を決定しなければならない。すなわち、1つの回避ポイントに向かうパスと次の回避ポイントに向かうパスを滑らかに連結できるようなパスを生成しなければならない [Kubota87c]。

ここでは、Figure 3.16 に示すように曲率半径一定のカーブを描いた曲線に沿って移動ロボットを誘導する。すなわち、移動ロボットからの回避ポイントの位置とその回避ポイントからサブゴールへの方向とから衝突回避開始地点、及び回避パスを計算する。

図中の  $x, \varphi$  は次式で計算される。ただし、 $R$  は実験的に求めている。

$$x = R \sin\left(\frac{\pi}{2} - \rho\right) + \sqrt{4R^2 - (R + R \cos\left(\frac{\pi}{2} - \rho\right) - \Delta)^2} \quad (3.8)$$

$$\varphi = \cos^{-1} \frac{R + R \cos\left(\frac{\pi}{2} - \rho\right) - \Delta}{2R} \quad (3.9)$$

ただし、

- $R$  : 移動ロボットがカーブする際の経路の曲率半径
- $\Delta$  : 回避幅
- $\rho$  : 回避ポイントAPにおけるサブゴールの方向

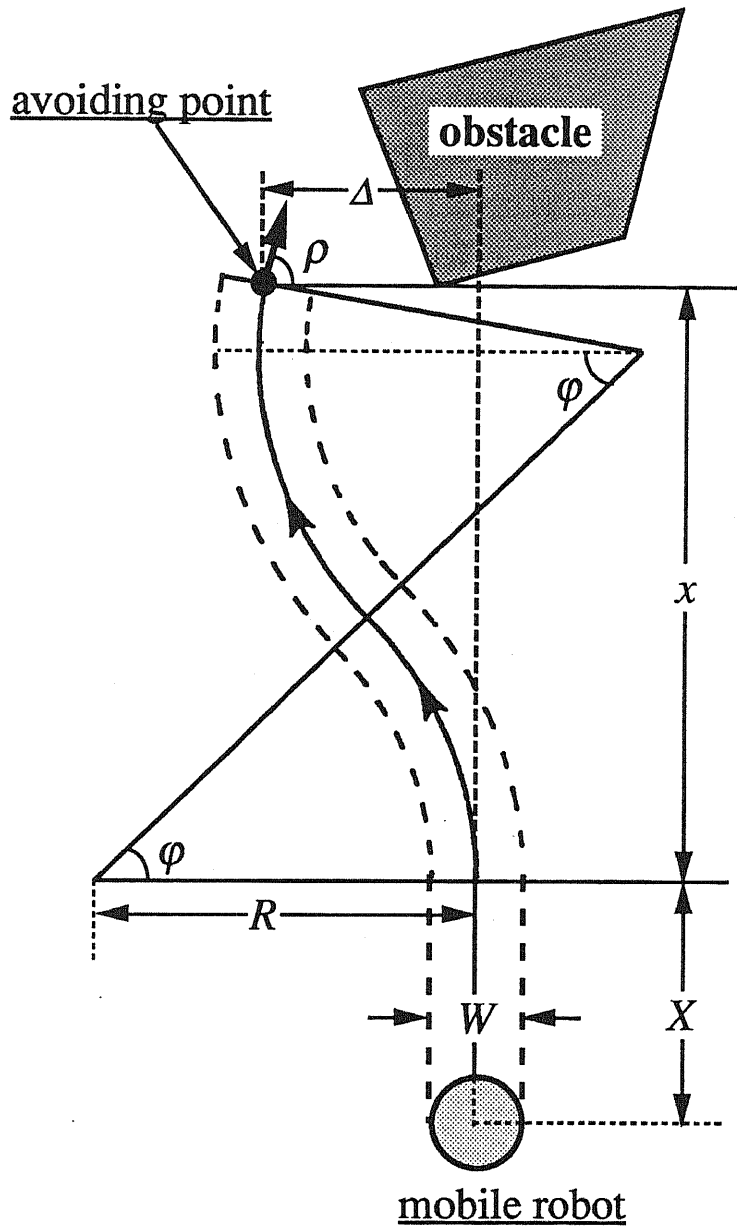


Figure 3.16 Navigation method

### 3.8 3次元グラフィックスによるシミュレーション

ここでは、ローカルパスプランニング手法の有効性を検証するために、3次元グラフィックスを用いて移動ロボットが視覚センサによって取り込む画像をシミュレートする。3次元コンピュータ・グラフィックスが2次元コンピュータ・グラフィックスと大きく違う点は、まず第一に、データと計算の量が2次元のときとは比べものにならないほど多いということである。空間内に物体を設定するため、 $x$ ,  $y$ 座標のほかに $z$ 座標も必要になる。さらに、2次元の多角形の場合には頂点のつながりだけを考えれば図形を表示することができたが、3次元空間内の多面体では、頂点のつながりをどのように表すか、面をどう扱うか等も問題になってくる。このような位置関係の複雑さのために、データ量も計算量も大幅に増えてしまう。第2の相違点は、空間内で設定した物体を最終的には画面という2次元の平面に、なんらかの形で投影しなければならないという点である。そのため、2次元で行った変換だけでなく、3次元物体を2次元に投影するための変換が必要になる。さらに、このときいくつかの物体が同じところに投影されることにもなる。物体が複数ある場合はもちろん、1つの物体の中でさえも見える面と見えない面とがでてくる。したがって、視点から見える部分を適切に表示する、隠面・隠線処理も必要になってくる。

移動ロボットが視覚センサによって取り込む画像をシミュレートするために、移動ロボット用3次元グラフィックス・パッケージを作成した。一般に、画像は投影変換によってあらわされるので、対象をフレームモデルによって表し、それを投影変換することによって画像をシミュレートした。ここでは、C言語を用いることにし、点のデータは、*point* という構造体にして、メンバとして $x, y, z$ それぞれの値を記憶する。面のデータは、*surface* という構造体にして面の色を *col* に、面を構成する頂点の番号列を *pnum* に入れる。

3次元グラフィックス・パッケージを用いてローカルパスプランニングのシミュレーションを行った。ここでは、サブゴールの位置を既知とし、まず、障害物が2個存在する環境におけるシミュレーションを行い、その結果を Figure 3.17 に示す。また、Figure 3.18 はスタート地点からゴールまでの移動ロボットの動きを上からシミュレートしたものである。この図から、移動ロボットはうまく回避ポイントを生成することによって、障害物に衝突することなくゴールに到達していることがわかる。さらに、障害物の高さがカメラの高さに比べて高い場合と低い場合に関してシミュレートした結果を

Figure 3.19 に示す。これらの場合にも画像データから回避ポイントを生成することが可能であり、提案する回避ポイント生成アルゴリズムが有効であることがわかる。

次に、障害物が3個存在する環境におけるシミュレーションを行い、その結果を Figure 3.20、Figure 3.21 に示す。Figure 3.20(a) はロボットがサブゴールの方向を向いて得た画像である。この画像を2値化し、前述の方法にしたがって回避ポイントを生成した結果を Figure 3.20(b) と Figure 3.20(c) に示す。Figure 3.20(d) と Figure 3.20(e) は移動ロボットが生成された回避ポイントへ向かって移動しはじめたときと回避ポイントに到達したとき得られた画像である。また、Figure 3.20(g) は2番目の障害物に関する2値画像と回避ポイントを示しており、移動ロボットは以上のプロセスを繰り返しおこなってサブゴールに到達する。Figure 3.21 はスタート地点からサブゴールまでの移動ロボットの動きを上からシミュレートしたものである。この図から、移動ロボットはうまく回避ポイントを生成することによって、障害物に衝突することなくサブゴールに到達していることがわかる。

ローカルマップ上に有限個の障害物が存在し、各障害物間がすべて移動ロボットの大きさに比べて十分大きいという条件を満たす環境におけるシミュレーション結果を Figure 3.22 に示す。シミュレーションでは、障害物を回避するために回避ポイントが逐次生成され、ロボットはゴール（サブゴール）に到達している。

以上のシミュレーション結果から、環境があまり複雑でない場合（各障害物間がすべて移動ロボットの大きさに比べて十分大きいなどの条件下）においてはローカルパスプランニングが十分に有効であることがわかる。



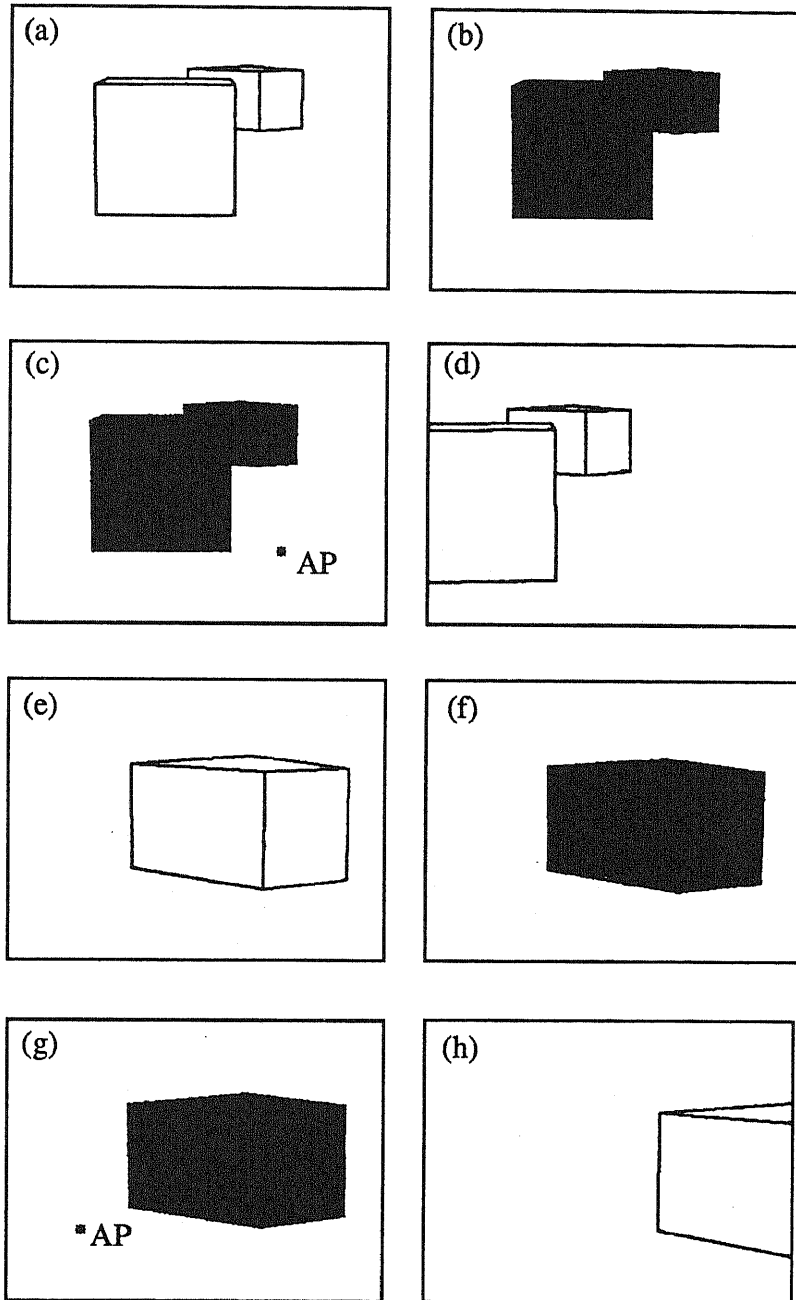


Figure 3.17 Simulation result 1-(1)

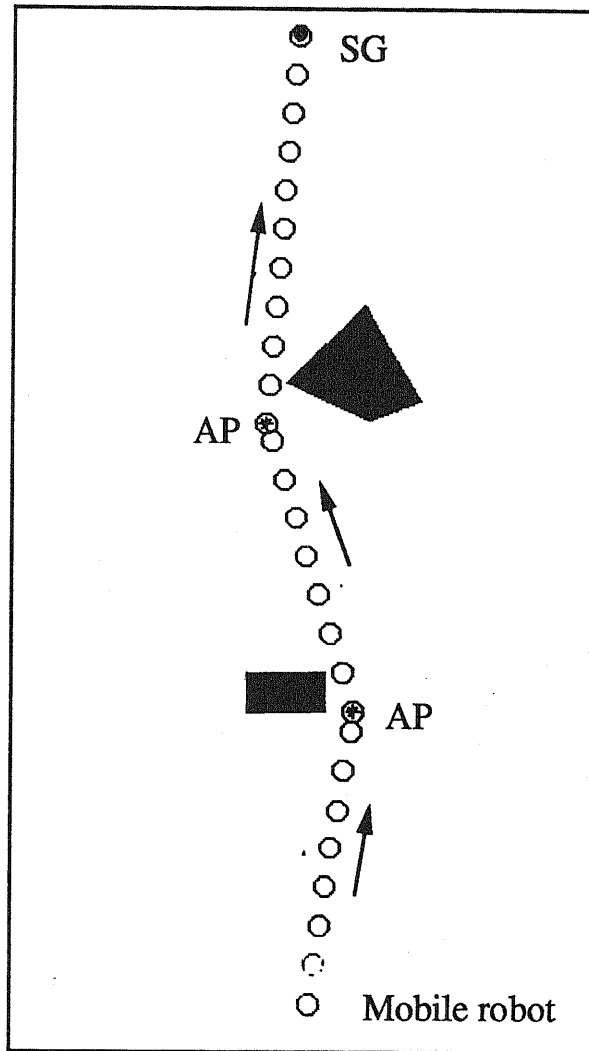


Figure 3.18 Simulation result 1-(2)

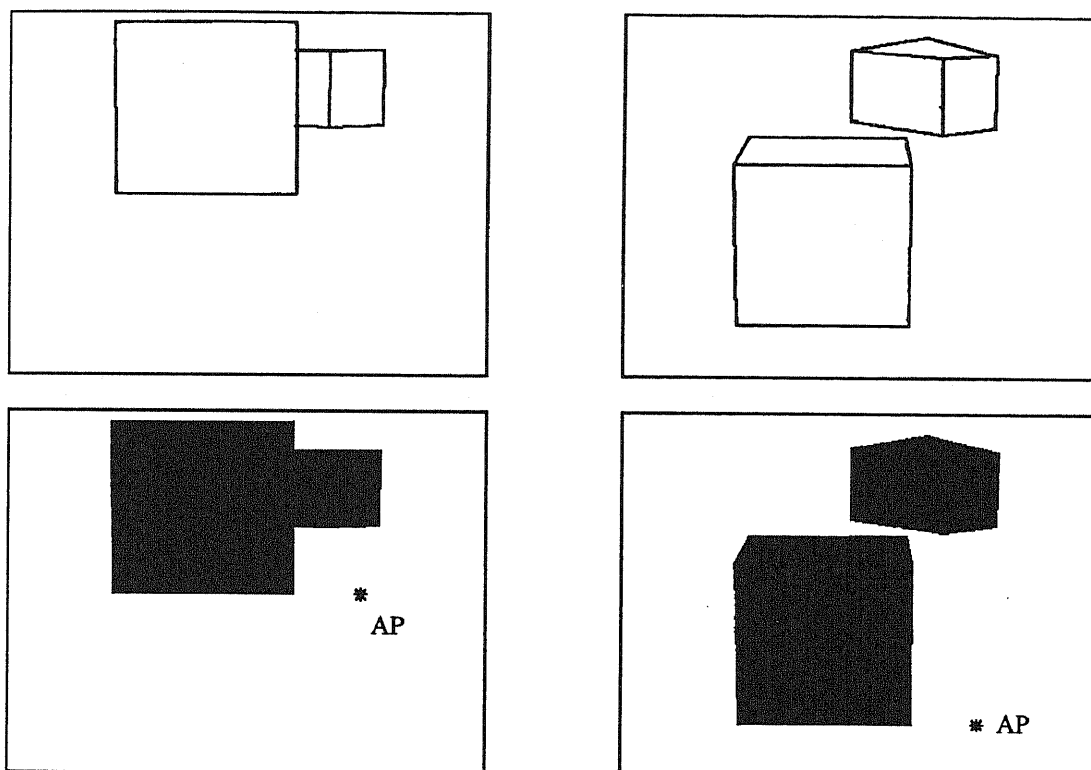


Figure 3.19 Simulation results 1-(3)

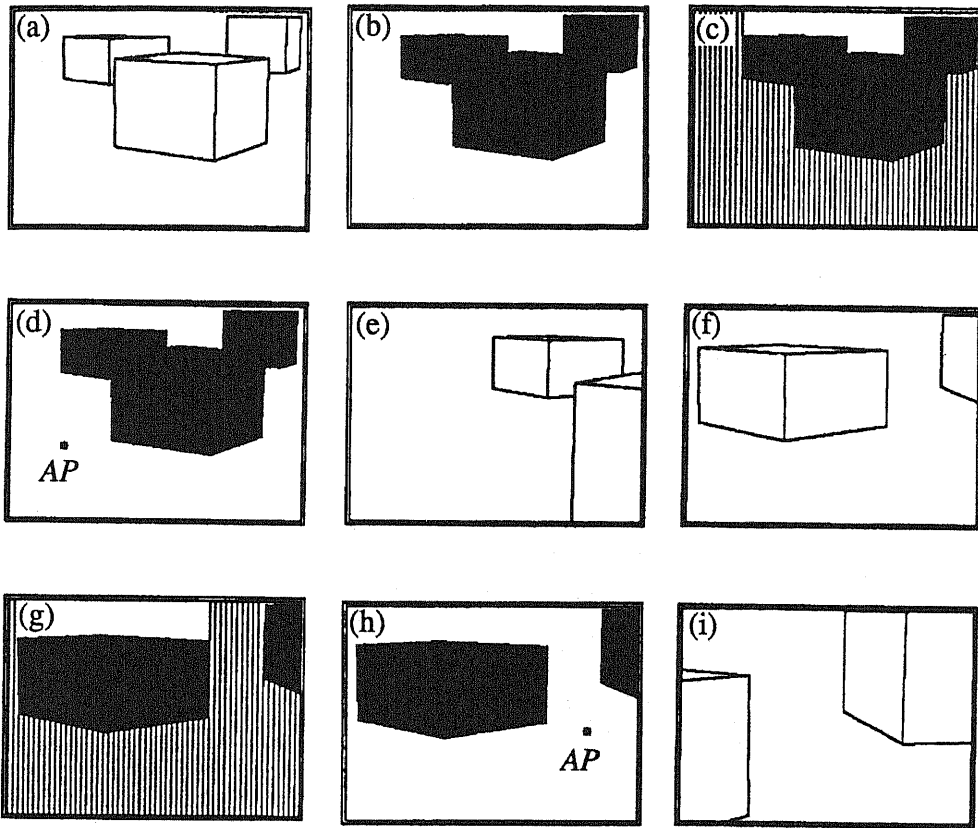


Figure 3.20 Simulation result 2-(1)

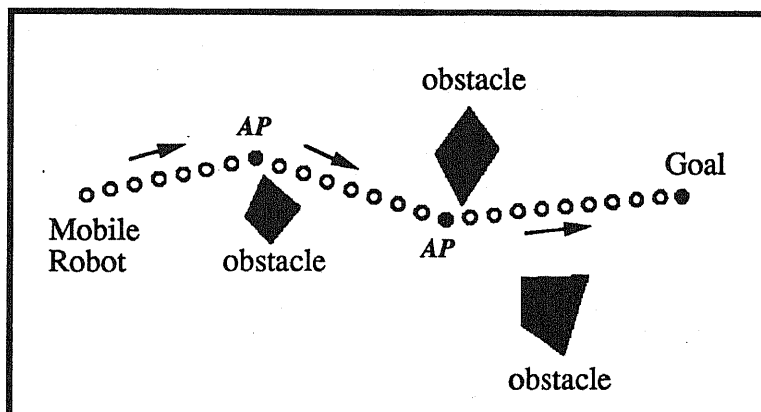


Figure 3.21 Simulation result 2-(2)

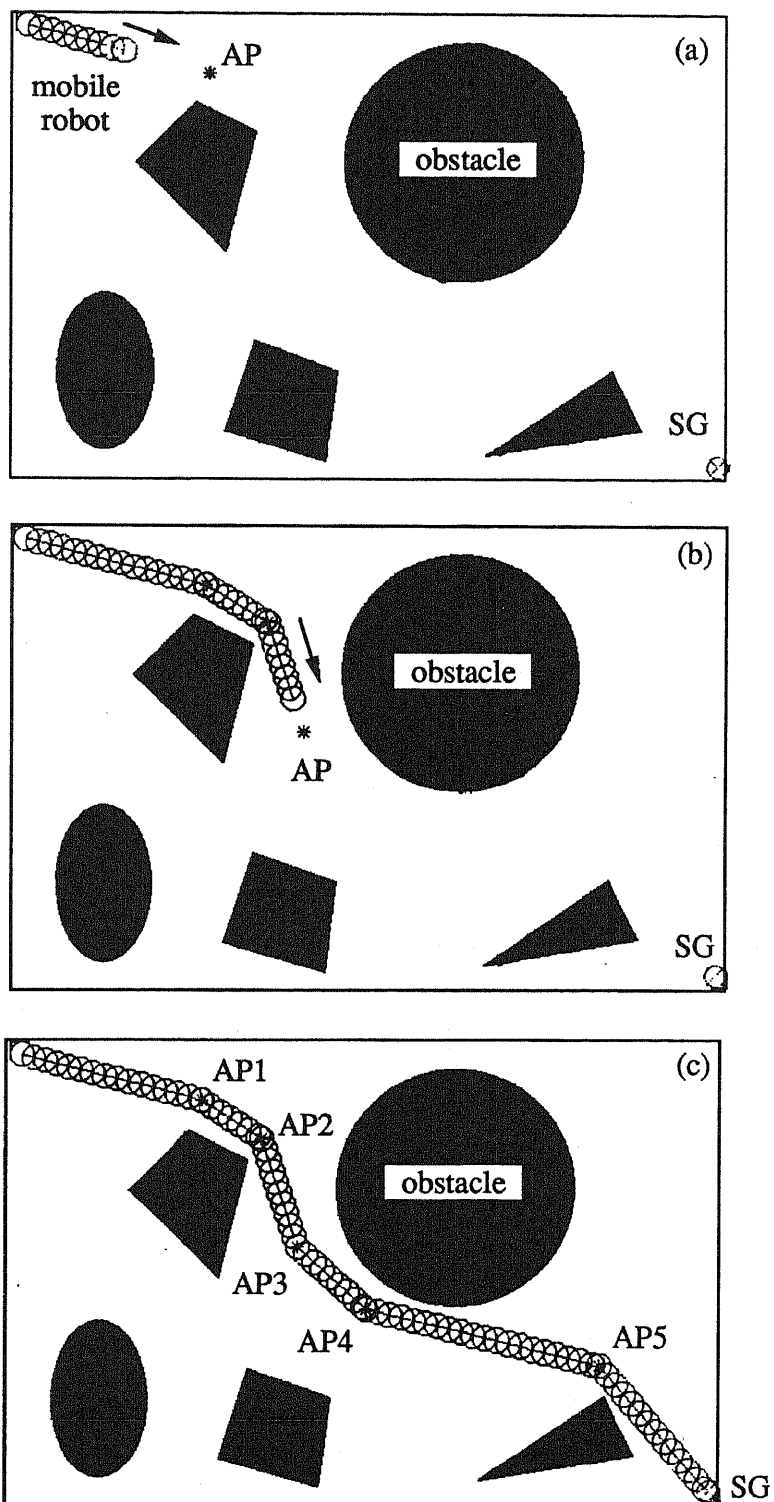


Figure 3.22 Simulation results 3

## 3.9 実験による検証

### 3.9.1 実験用移動ロボットシステム

実験に用いた移動ロボットは、主な構成要素として CCDカメラを搭載した移動ロボット本体と、画像処理、プランニング、ナビゲーション、コントロール等を行うマイクロコンピュータからなる。移動ロボットシステムの構成および外観図を Figure 3.23 及び Figure 3.24 に示す。

#### ○移動ロボット本体

移動ロボットの縦・横は共に約 15[cm]、高さは約 25[cm]、重量は約 2[kg] である。底面には一対のステッピングモーター駆動輪および一対のボールキャスターが取り付けられており、前後直進とその場回転とができる。駆動輪の制御は、ロボットに搭載された Z80 ボードとステッピングモーター駆動用のドライブ回路を介し、ホストコンピュータから制御パルスを送信することにより行う。Z80 ボードは、主にステッピングモーターの駆動パルスの発生に用いられ、ホストコンピュータと 7bit 平行通信をすることができる。また、ロボットの頭部には小型の CCDカメラ (SONYXC38) 1 台が取り付けられており、移動しながら画像を取り込むことができる。CCDカメラによって得られる画像信号は、NTSC信号としてホストコンピュータのイメージメモリに送られ、デジタル化されてメモリされる。この CCDカメラは外部同期型で、イメージメモリのフリーズ信号に同期して画像を取り込む。モーター駆動用電源と IC用電源は、ロボット本体の外におき、ケーブルを通じてロボット本体と結合している。

#### ○イメージメモリボード

CCDカメラによって取り込まれた画像は、ホストコンピュータに取り付けられたイメージメモリボード (ALTEC ALT-256-8) を通じて、解像度 256 × 256、輝度レベル 8bit の濃淡画像データに変換され、ホストコンピュータの VRAM上へ一画像当り、約 0.03sec 以内に DMA 転送される。マイクロコンピュータは、VRAM のデータをアクセスすることによって画像データを扱うことができる。また、モニタ用に白黒 TV モニタが、メモリボードのビデオ出力端子に接続されており、画像データをモニタすることができる。

### ○マイクロコンピュータ

マイクロコンピュータ (NEC PC-9801 VM2 プロセッサ V30) は、画像処理、プランニングおよびコントロール等を行う。ホストコンピュータには、移動ロボットからの画像信号 (NTSC信号)、移動ロボット制御用の 7bit パラレル通信用ケーブルが接続されている。ロボットの制御計画は、マイクロコンピュータ上でC言語によりプログラムされている。

各システムの構成図を Figure 3.25 に示す。

### 3.9.2 未知障害物回避実験

ここでは、ゴール (あるいはグローバルプラナーから与えられたサブゴールと考えるてもよい) の位置を既知とし、また実験装置の制約から、障害物が2個存在する場合について実験を行った。本実験において、第 3.5.2 節で述べた回避ポイントの生成方法に従って、エッジ検出関数 *search* によって得られたエッジ情報から回避ポイントを生成している。ただし、エッジ検出アルゴリズムにおいて、 $\delta = 8$ 、しきい値 = 10 という値を用いた。実験のおおよそのようすを Figure 3.26 に示す。

移動ロボットは、障害物に衝突することなく、ゴールに到達することができた。このとき、移動ロボットの速度は約 13[cm/s] で、画像処理時間は約 0.6[s] であった。

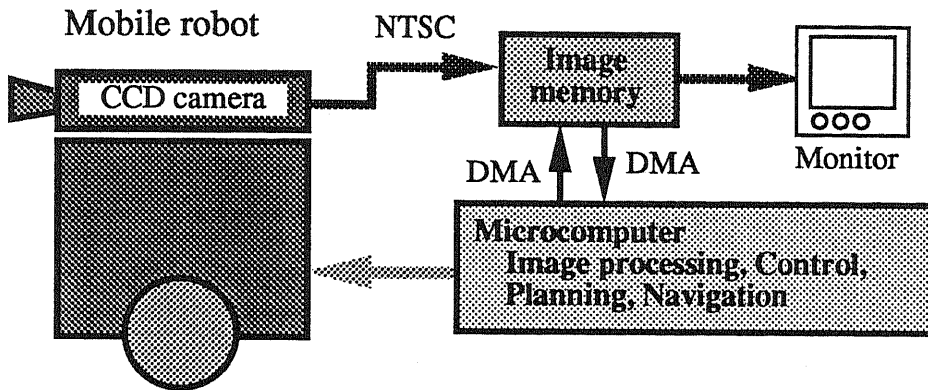


Figure 3.23 Configuration experimental system

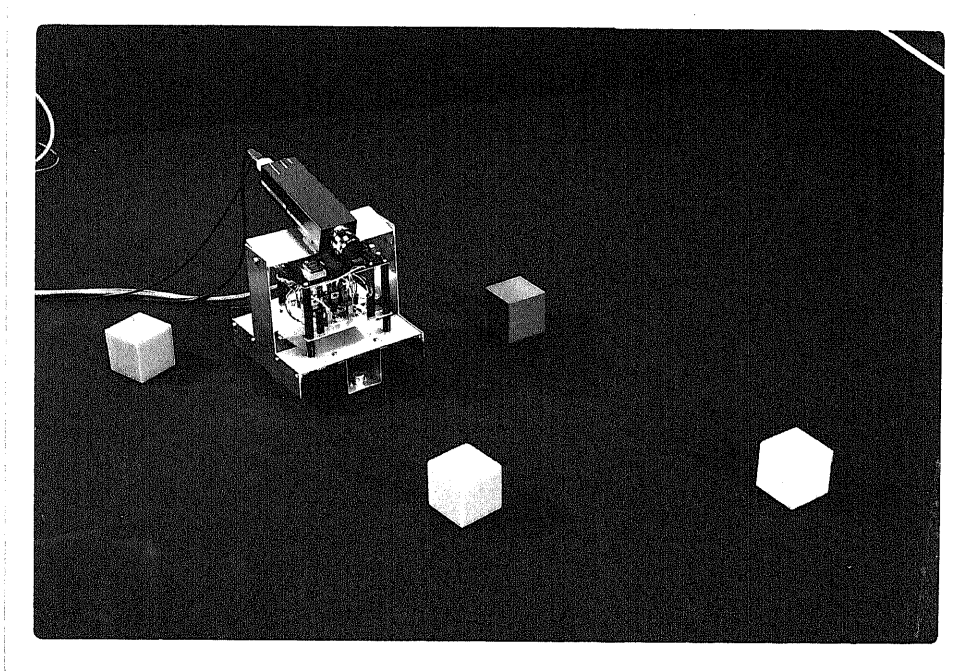


Figure 3.24 Overview of the mobile robot



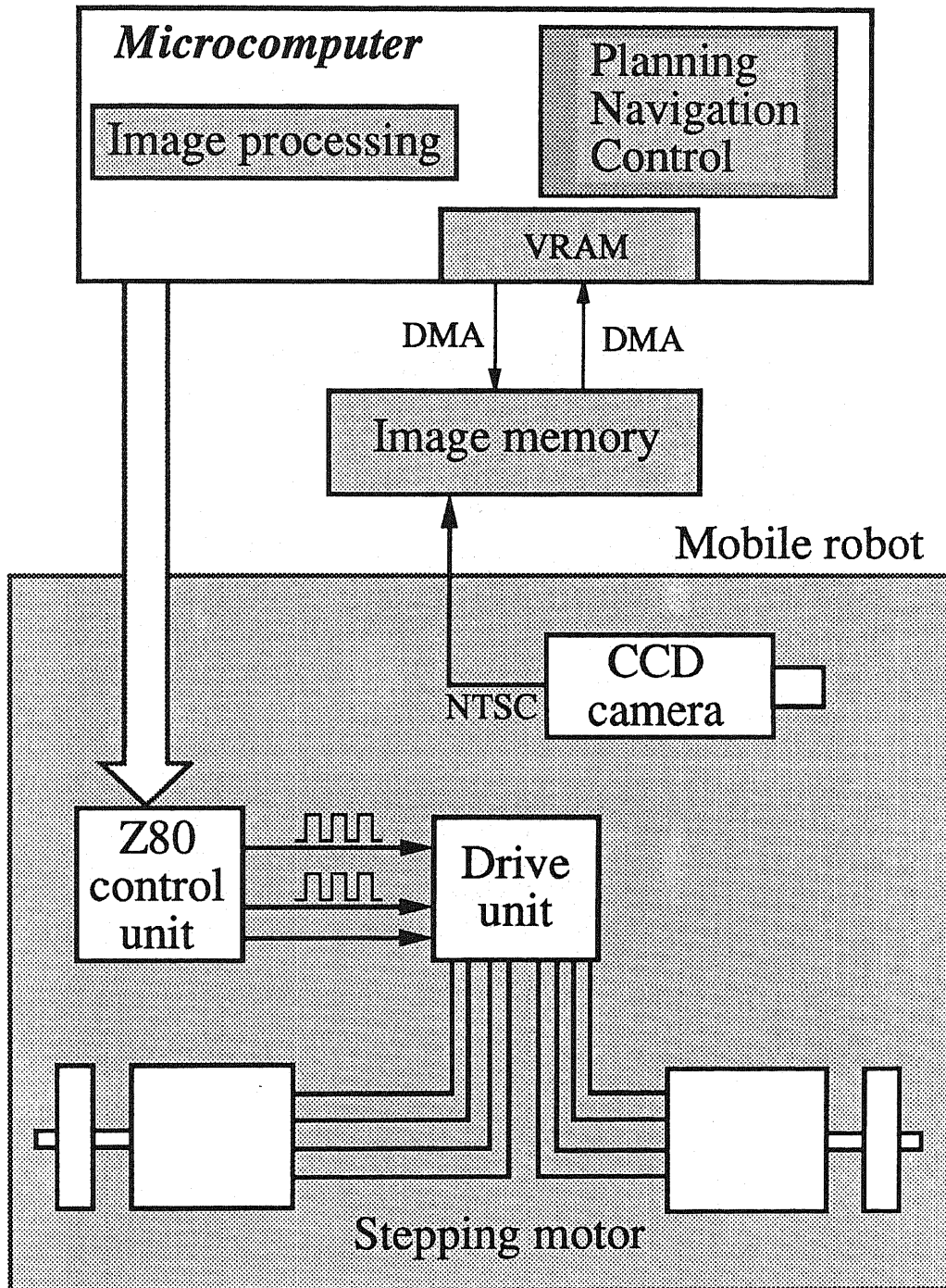


Figure 3.25 Mobile robot control system

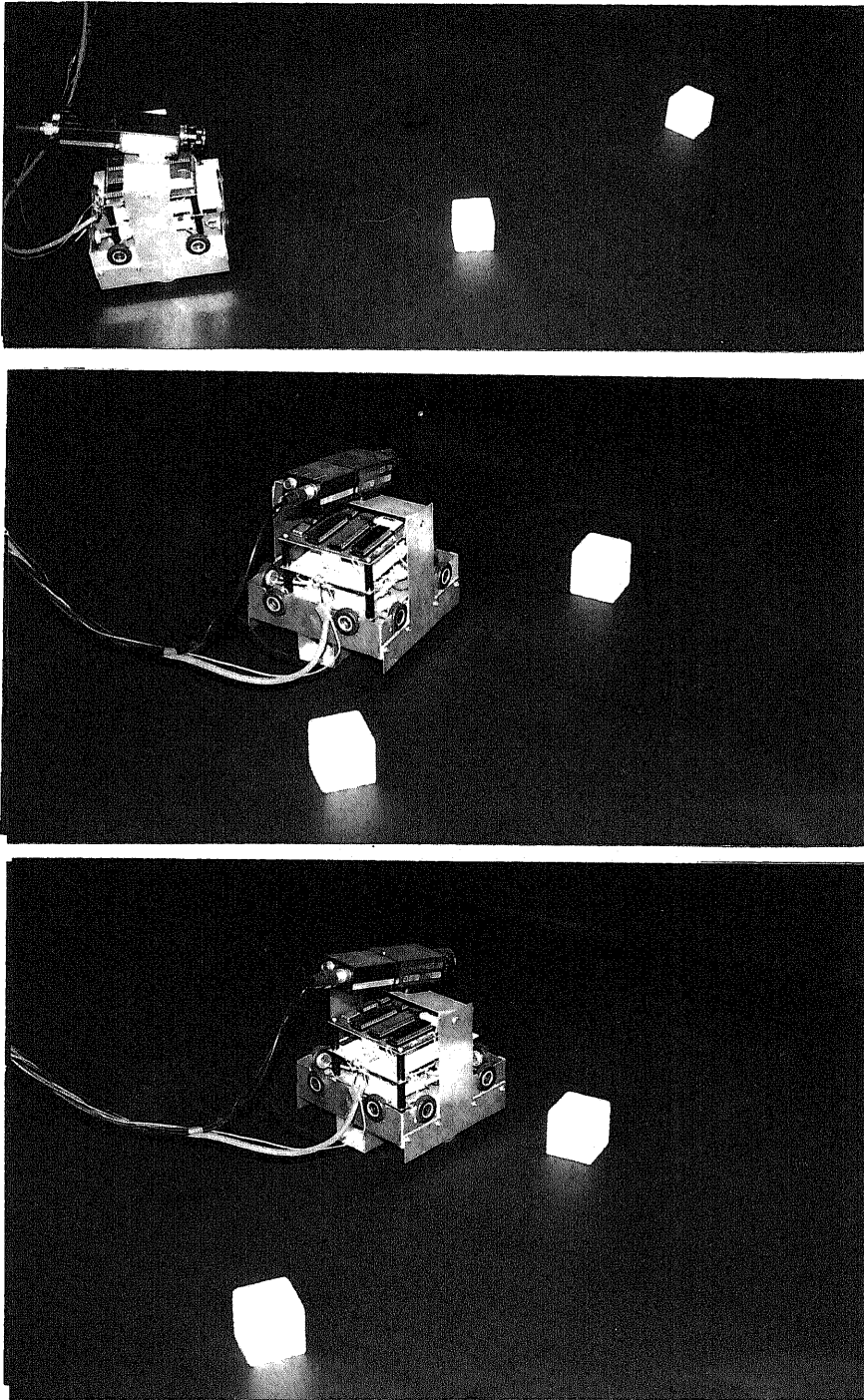


Figure 3.26 Experimental result

### 3.10 まとめ

あいまいな環境において有効な経路探索方法として、階層的プランニング方法を提案した。これはスタートからゴールまでの経路をサブゴールマップの概念を用いて決定するグローバルパスプランニングと、障害物回避などの実際に移動しながらロボットの環境に応じて行うローカルパスプランニングとからなる。この方法は環境地図とセンサ情報を有効に利用できるという特徴をもつ。また、ローカルプランニングにおいて画像データから回避ポイントの生成を試み、シミュレーションによってその有効性がわかった。問題点としては、グローバルプランニングにおいて従来の方法を利用したサブゴールマップ生成の概念を述べたが、人間が与える大域的な地図の表現方法妥当性の評価、実際のサブゴールマップの生成法、どのような評価関数で経路を決めるかなどの問題が残っている。

実験用移動ロボットを用いて、ローカルパスプランニング手法の実験的検証を行った。実験では障害物が複数ある場合の画像処理アルゴリズムを用いており、提案したアルゴリズムが有効であることがわかった。

## 参考文献

- [Barr81] A.Barr, E.A.Feigenbaum, "The Handbook of Artificial Intelligence," 1, William Kaufman (1981).
- [Brooks85] R.A.Brooks, "A Robust Layered Control System for a Mobile Robot," IEEE Journal of Robotics and Automation, Vol. RA-2, No.1 (1985).
- [原島86] 原島, 山本, 橋本, "視覚センサを用いた移動ロボットの障害物回避," 第4回日本ロボット学会学術講演会予稿集, pp.17-18 (1986).
- [Harashima87] F.Harashima, H.Yamamoto, H.Hashimoto, "Obstacle Avoidance Using Visual Information for Mobile Robot," IEEE Int. Workshop on Industrial Application of Machine Vision and Machine Intelligence, pp.272-277 (1987).
- [市川84] 市川, 妹尾, 宮田, "移動機械の自律制御," 計測自動制御学会論文集, 第20巻第10号, pp.87-91 (1984).
- [Isik86] C.Isik, A.Meystel, "Decision Making at a Level of a Hierarchical Control for Unmanned Robot," Proc. of IEEE Int. Conf. on Robotics and Automation, Vol.3, pp.1772-1778 (1986).
- [柿倉86] 柿倉, "移動ロボットの経路探索," 日本機械学会誌, 89-815, pp.1151-1156 (1986).
- [柿倉87] 柿倉, "自律移動のロボットにおける知能化技術(1) 行動プランニング," 日本ロボット学会誌5巻5号, pp.66-70 (1987).
- [Koch85] E.Koch, C.Yeh, G.Hillel, A.Meystel, C.Isik, "Simulation of Path Planning for a System with Vision and Map Updating," Proc. of IEEE Int. Conf. on Robotics and Automation, pp.146-160 (1985).
- [久保田87a] 久保田, 橋本, 原島, "移動ロボットにおけるセンサ情報処理," 第26回計測自動制御学会学術講演会予稿集, No.JS49-2 (1987).
- [久保田87b] 久保田, 橋本, 原島, "階層的プランニングによる経路探索(実験的検証)," 第5回日本ロボット学会学術講演会予稿集, pp.349-352 (1987).
- [Kubota87c] T.Kubota, H.Hashimoto, F.Harashima, "Control of Mobile Robot Based on Cooperation of Vision and Range Sensors," Proc. of Int. Workshop on Robotics : Trends, Technology and Applications, pp.197-206 (1987).

- [久保田 88a] 久保田, 橋本, 原島, "単眼視による移動ロボットの複数障害物回避 (回避ポイント生成のための画像処理アルゴリズム)," 第27回計測自動制御学会学術講演会予稿集, No.JS37-3 (1988).
- [Kubota88b] H.Hashimoto, T.Kubota, F.Harashima, "Path Planning of Mobile Robot Based on Visual Information," Proc. of USA-JAPAN Symposium on Flexible Automation, pp.637-644 (1988).
- [久保田 89a] 久保田, 橋本, 原島, "視覚センサを用いた移動ロボットのローカルパスプランニング," 第28回計測自動制御学会学術講演会予稿集, No.JS3-3 (1989).
- [久保田 89b] 久保田, 橋本, 原島, "ローカルプランニングによる移動ロボットの経路探索," 日本ロボット学会誌, Vol.7, No.4, pp.3-10 (1989).
- [松本 87] 松本, 油田, "経路地図に従った移動ロボットの自律走行システム," 日本ロボット学会誌 5 巻 5 号, pp.19-28 (1987).
- [Moravec83] H.P.Moravec, "The Stanford Cart and the CMU Rover," Proc. of IEEE, Vol.71, No.7, pp.872-884 (1983).
- [Nilson69] N.J.Nilson, "A Mobile Automation : An Application of Artificial Intelligence Techniques," Proc. of 1st IJCAI, pp.509-520 (1969).
- [登尾 89] 登尾, 浪花, 有本, "クワッドツリーを利用した移動ロボットの高速経路生成アルゴリズム," 日本ロボット学会誌, Vol.7, No.5, pp.403-413 (1989).
- [Raviv87] D.Raviv, Y.H.Pao, K.Loparo, "Reconstruction of Three Dimensional Surfaces from Two Dimensional Binary Images," Int. Workshop on Robotics : Trends, Technology and Applications, pp.185-193 (1987).
- [白井] 白井, "コンピュータビジョン," 昭晃堂.
- [Shirai83] Y.Shirai, "Recognition of Polyhedrons with Range Finder," Pattern Recognition 4, pp.872-884 (1983).
- [Tachi84] S.Tachi, K.Komoriya, "Guide Dog Robot," Proc. of 2nd Int. Symp. of Robotics Research, pp.333-340 (1984).
- [Tsuji85] S.Tsuji, Y.Yagi, M.Asada, "Dynamic Scene Analysis for a Mobile Robot in a Man-Made Environment," Proc. of IEEE Int Conf. on Robotics and Automation, pp.850-855 (1985).

[Wallance86] R.Wallance, et al., "Progress in Robot Road-Following," Proc. of IEEE Int. Conf. on Robotics and Automation, pp.1615-1621 (1986).

## Chapter 4

# センサの協調による 移動ロボットのナビゲーション

本章では、まず単眼視による環境観測を行い、視覚情報を用いた移動ロボットのナビゲーション法について述べる。単眼視による環境の理解は、必ずあいまいさを伴う。そこで、このあいまいさを減少させるために、超音波センサ等の近接センサを同時に用いる。ここでは、ロボットが必要な情報を信頼性高く獲得するためにセンサ情報の融合を試みる。

## 4.1 はじめに

ロボットが現実の世界をとりあつかうためには、必要な情報を信頼性高く獲得することが必要である。環境を認識し、それに適応して行動する知能移動ロボットにおいて、視覚センサの果たす役割は大きい。そこで、視覚情報を用いた移動ロボットの誘導方法について考える。ロボットの感覚機能として一種類のセンサでは正確な情報を獲得することはなかなかできず、また各センサには得意とする情報と苦手とする情報とがある。したがって、各種センサから取得した情報を統合・融合することによって、行動の決定に役立つ信頼性の高い情報を獲得することができると考えられる。

移動ロボットにおいて、視覚センサは大域的な情報は得やすいが、正確な近接情報を得ることは難しい。そこで、超音波センサ等のレンジセンサを用いて近接距離情報を融合して、より正確なコントロールを行うための情報を作らなければならない。センサ情報の融合に関する研究としては、一種類のセンサが複数個ある場合に、そのデータを補正する方法 [Luo88] や黒板システムなどを用いて環境認識などの上位レベルでのセンサ情報を融合する方法 [Harmon86] などがある。しかし、これらの方法は、補正を行ったり、単なるセンサ情報の組合せであって、センサからの情報を融合して新しい情報を得た研究はあまりない。また、コントロール情報などのリアルタイム性を要求されるレベルでのセンサ情報融合に関する研究も少ない。そこで、移動ロボットのコントロールという観点から、センサ情報の融合を考えてみる。

本章では、まず単眼視による環境観測を行い、視覚情報を用いた移動ロボットのナビゲーション法について述べる。単眼視による環境の理解は、必ずあいまいさを伴う。そこで、このあいまいさを減少させるために、超音波センサ等の近接センサを同時に用いる。ここでは、ロボットが必要な情報を信頼性高く獲得するためにセンサ情報の融合を試みる。センサ情報処理として、センサ情報のデータ構造、処理手順について述べる。実際に視覚センサ情報とレンジセンサ情報の融合方法を提案し、シミュレーションによってその有効性を示す。さらに、実験用移動ロボットを用いて、視覚センサと超音波センサの協調による実験を行い、本手法の実験的検証を行う。



## 4.2 移動ロボットのビジュアルナビゲーション

移動ロボットは、与えられた環境のもとで初期点から出発して、障害物を回避しながら目標点に到達する。本研究では、屋内環境を走行する移動ロボットを視覚情報を用いて与えられた目標に誘導する方法について考える。

### 4.2.1 問題設定

本研究では、作業空間として屋内を想定し、単眼視による環境観測を行う。一般に、静止した1台のカメラ画像から、奥行き情報は得られないが、「環境内の物体は床面上に立っていて、その最下部が見えている」という発見的知識を用いて、物体の位置や大きさを算定する。前提条件として、あらかじめサブゴールマップが与えられているものとする。第3章で述べたように、サブゴールマップは視覚的に確認し、追従できる目標（サブゴール）の集合からなり、各サブゴールは、その視覚的特徴、および隣接するサブゴールの方向や距離などの情報をもっている。ここでは、スタートとゴールを結ぶパスがサブゴールのシーケンスという形で与えられているものとし、視覚センサを用いて移動ロボットを与えられたサブゴールに誘導することを考える。また、ロボットが移動中に視覚情報を用いてフィードバック制御を行い、リアルタイム処理を行うという観点から、サブゴールは床面上にあり、まわりの環境と何等かの色、明暗の変化があるものと仮定する。

## 4.3 目標追従動作

### 4.3.1 視覚情報の利用

視覚情報を制御ループの中に取り入れたビジュアルフィードバックに関する研究が盛んに行われてきている。移動ロボットにおいても、ビジュアルフィードバックを用いて視覚誘導しようとする研究 [南方 86][亀島 87] が見られる。しかし、移動ロボットの位置や姿勢を視覚によって決定しようとするために処理時間が問題となり、実際のコントロール法まで言及している研究は少ない。ALV (Autonomous Land Vehicle) の研究の一環として、カメラによって道路のセンタを検出してALVを道路に沿うようにステアリングする方法 [Wallance85] も報告されている。この方法では、道路のセンタのような直線上の目標に沿うようなステアリングコントロールを行っている。このため、一定の水平方向の直線上における画面のセンタと道路のセンタとの偏差をそのままフィードバックしており、目標が一点の場合には目標までの距離によってフィードバックゲインを調整する必要がある。ここでは、ポイント状の目標に追従する方法について検討する。

### 4.3.2 ビジュアルナビゲーション

本研究では、ロボットが移動中に視覚情報を用いてフィードバック制御を行い、リアルタイム処理を行うという観点から、問題設定で述べたようにサブゴールは床面上にあり、まわりの環境と何等かの色、明暗の変化があるものとする。サブゴールが視覚センサによって検出できる場合、その目標に直進するようにコントロールすればよい。まず最初に画像を視覚センサによって取り込み、目標の画像面上の位置を検出する。カメラが移動ロボットの進行方向に固定されているとすると、画像面の中央の鉛直方向の直線に一致するように移動ロボットの移動をコントロールしてやればよい。このコントロールはおもに移動ロボットのステアリング調整によって行われる。

サブゴールの位置情報は、ロボットの現在位置を中心に設定した2次元座標系内のベクトルで与えられる。また、これは視覚センサから得られた画像上の点としても与えられる。Figure 4.1(a) は移動ロボットが視覚センサから取り入れた画像を示している。画像上の\*は与えられたサブゴール (SG) を表す。ロボットは画像のサンプリングごとにサブゴールの位置を検出し、サブゴールと画像のセンターラインとの偏

差  $d$  とサブゴールの鉛直方向での位置  $h$  の値を得る。Figure 4.1(b) は、移動ロボットとサブゴールを上からみた図を示している。ロボット座標系は、移動ロボットの位置を原点とし、移動ロボットの進行方向が  $Y$  軸となるように設定している。図中の  $\alpha$  は移動ロボットの進行方向とサブゴールの方向とがなす角である。この角  $\alpha$  をフィードバックすることにより、ロボットをサブゴールに誘導する。 $\alpha$  は画像から検出された  $d$  と  $h$  から次のようにして計算される。

いま、Figure 4.1(c) における  $\beta$  は次式で表される。

$$\beta = \beta_0 + \gamma \quad (4.1)$$

$$\gamma = \tan^{-1} \frac{h}{F} \quad (4.2)$$

また、画像上で検出されるセンターラインからの偏差  $d$  は次式で表される。

$$d = \frac{F}{L \cos \gamma + F} RX \quad (4.3)$$

ただし、

- $\beta_0$  : カメラの傾斜角
- $F$  : 焦点距離
- $L$  : 焦点から対象までの距離 (光軸上)
- $RX$  : 移動ロボットの進行方向線分とサブゴールの距離

これらの式を用いると

$$RX = \frac{dH \cos \gamma}{F \sin \beta} \quad (4.4)$$

$$RY = H \cot \beta \quad (4.5)$$

ただし、

$Y$  : 移動ロボットからサブゴールまでの距離の進行方向成分

$H$  : カメラの床面からの高さ

などの関係式が得られる。

ところで、角  $\alpha$  は次の式で表される。

$$\alpha = \tan^{-1} \frac{RX}{RY} \quad (4.6)$$

式(4.4)、式(4.5)を代入して

$$\alpha = \tan^{-1} \frac{d \cos \gamma}{F \cos \beta} \quad (4.7)$$

さらに、式(4.2)を代入すると次のようになる。

$$\alpha = \tan^{-1} \frac{d \cos(\tan^{-1} \frac{h}{F})}{F \cos \beta} \quad (4.8)$$

この  $\alpha$  を用いて、移動ロボットの姿勢を次の式で表すことにする。

$$\frac{d\theta}{dt} = K\alpha \quad (K : \text{constant}) \quad (4.9)$$

ゲイン  $K$  を適当に選ぶことにより、移動ロボットをサブゴールに誘導することができる。なお、ここでは移動ロボットのパス生成を主眼としているので、速度制御は考慮していない。

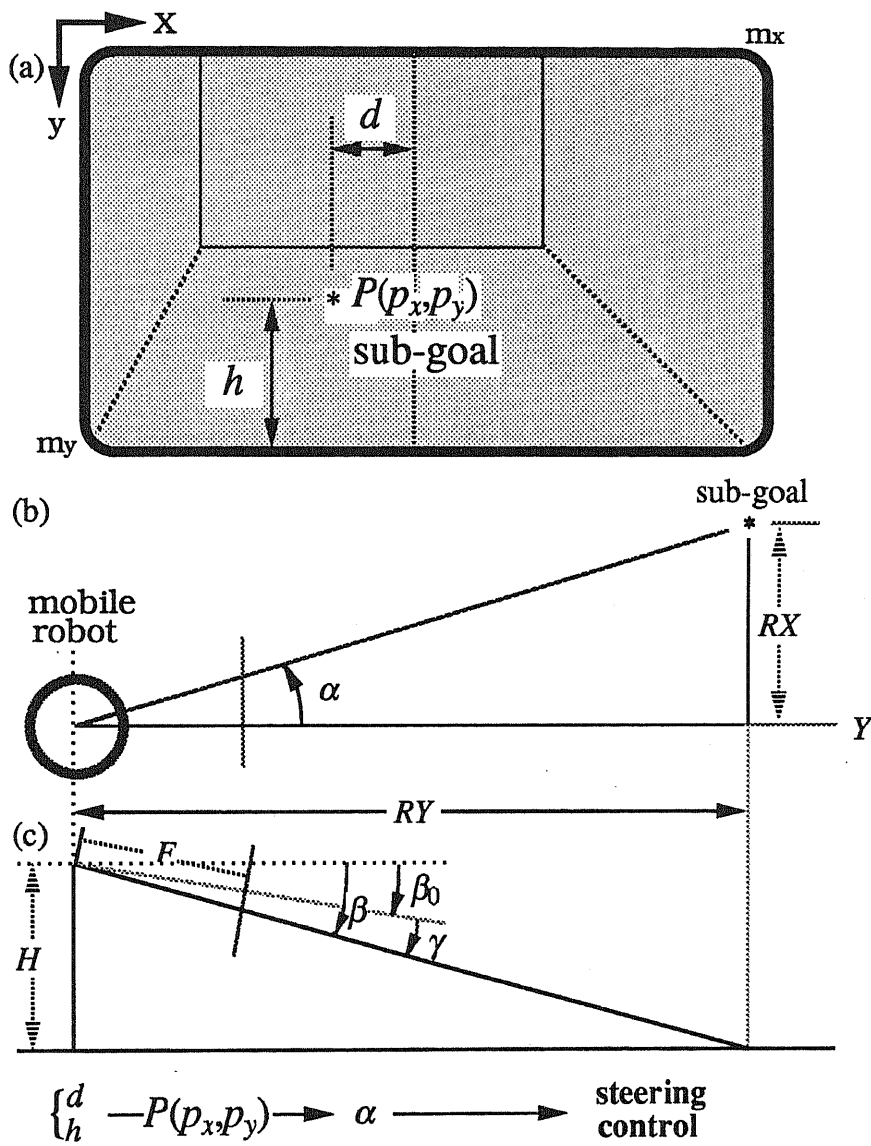


Figure 4.1 Visual following method

### 4.3.3 シミュレーションによる考察

移動ロボットの視覚目標追従シミュレーション結果を Figure 4.2 に示す。小四角形が移動ロボットの動きを表し、プラス記号 (+) が目標を表している。ここでは、比例ゲインを変化させたときの追従動作の変化をシミュレートしている。この結果より、移動ロボットの動きは2次系に近い動きをしていることがわかる。ゲイン  $K$  としては、 $K = 1.0$  程度が臨界振動に近い振舞いを示している。いま、このシステムを次のように記述する。

$$\frac{d\theta}{dt} = K\alpha \quad (K : \text{constant}) \quad (4.10)$$

$$\frac{dp}{dt} = -v \cos \alpha \quad (4.11)$$

$$\frac{d\theta}{dt} + \frac{d\alpha}{dt} = \frac{v \sin \alpha}{p} \quad (4.12)$$

ただし、

- $p$  : 移動ロボットから目標までの距離
- $v$  : 移動ロボットの速さ

式(4.10)を式(4.12)に代入すると

$$\frac{d\alpha}{dt} = \frac{v \sin \alpha}{p} - K\alpha \quad (4.13)$$

これより、

$$p = \frac{v \sin \alpha}{\frac{d\alpha}{dt} + K\alpha} \quad (4.14)$$

式(4.14)を式(4.11)に代入して整理すると次のようになる。

$$v \sin \alpha \left( \frac{d^2 \alpha}{dt^2} + K \frac{d\alpha}{dt} \right) = v \cos \alpha \left\{ \left( \frac{d\alpha}{dt} + K\alpha \right) + \left( \frac{d\alpha}{dt} + K\alpha \right)^2 \right\} \quad (4.15)$$

ここで、 $\alpha \approx 0$  とすると、

$$\alpha \frac{d^2 \alpha}{dt^2} - \left( \frac{\alpha}{dt} \right)^2 - K\alpha \frac{d\alpha}{dt} - \frac{d\alpha}{dt} - K\alpha = 0 \quad (4.16)$$

となる。

しかしながら、解析的に解くことは難しいため、ここでは次の定性的解釈を簡単に記す [山本 87]。Figure 4.3 は、 $K \cdot \Delta T = 2.0$  ( $\Delta T$  はサンプリングタイム) のときのワンサンプリング間の移動ロボットのとり経路を示す。図に示すように、この場合観測される目標との偏差角  $\alpha$  の大きさは、ワンサンプリングの移動の後も変わらない。すなわち、振動的な移動を繰り返すことになる。

ワンサンプリング後に観測される偏差角  $\alpha$  は、移動ロボットが自分の中心に対して回転することによる作用と、目標に対する移動ロボットの中心の回転による作用の重ね合わせによって決まる。したがって、ゲインが小さい場合でも、偏差角  $\alpha$  が 0 に収束するよりもはやく目標に近づきすぎると、目標に対する回転による作用が大きくなり、偏差角  $\alpha$  は発散する。

以上のことより、最小ゲイン  $K$  は、目標までの距離とワンサンプリング間の移動距離 (速度) によって決まる。目標までの距離が移動ロボットの速度に対して十分大きい場合には、移動ロボットのゲインは  $K \cdot \Delta T = 2.0$  より小さければ、偏差角  $\alpha$  は収束し、移動ロボットは目標に到達できると考えられる。

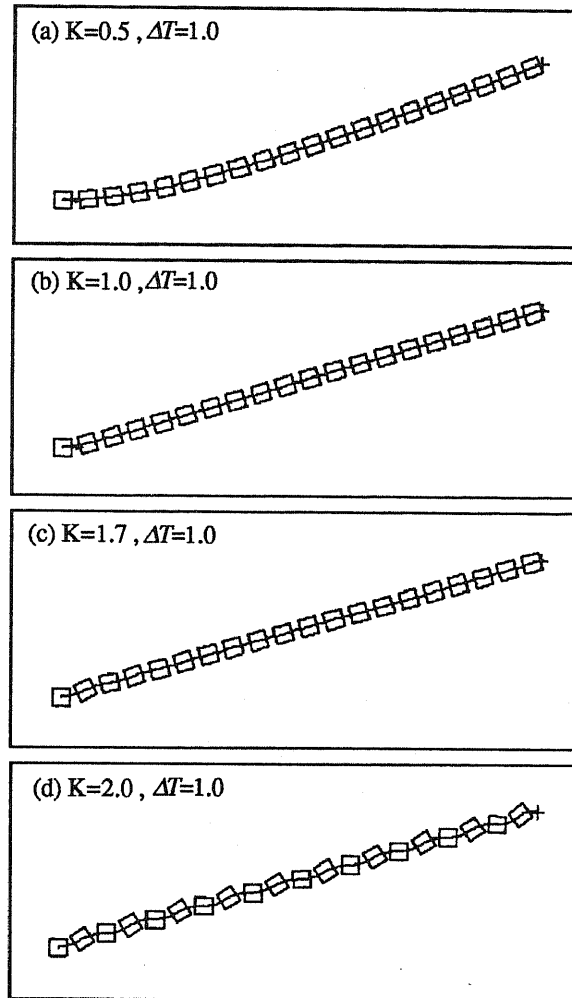


Figure 4.2 Simulation results

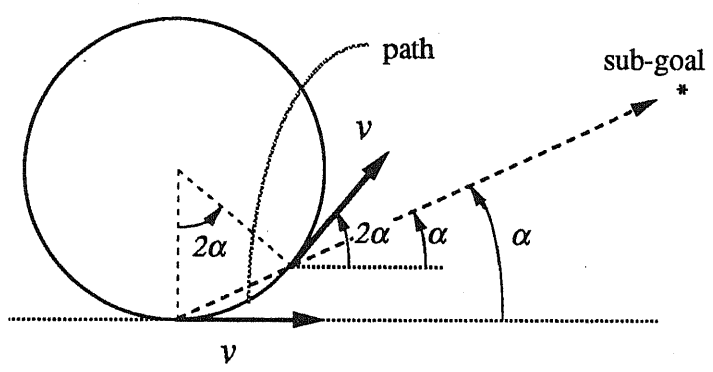


Figure 4.3 Path within one sampling period



## 4.4 障害物回避

移動ロボットを視覚センサを用いてナビゲーションする場合に問題となるのは、経路上に未知の障害物などがあってサブゴールを確認できない場合である。このような場合には、視覚情報から障害物の位置を検出し、到達可能なサブゴール (*new sub-goal*) を新たに生成する。移動ロボットは障害物に会うたびごと、*new sub-goal* を繰り返して生成して障害物回避を行い、本来のサブゴールに到達することができる。

### 4.4.1 回避経路の生成

障害物を回避するために、まず障害物の左右のどちらかにサブゴールを設ける。ここでは、視覚センサから得た画像において障害物の左右エッジのうち、画像の中心線にどちらが近いかを考え、障害物のどちら側に回避するかを決定する。エッジ点と中心線との距離、移動ロボットの幅などを考慮してサブゴールの位置を決定する。次に現在地点と新たなサブゴールを結ぶ障害物回避経路を生成する。このとき、得られる情報はその新たなサブゴールの位置情報だけではなく、その点における次のサブゴールの位置情報も含まれる。したがって、到達すべき新たなサブゴールとその点において移動ロボットが持つべき速度 (姿勢) を考慮して経路を決定しなければならない。すなわち、1つのサブゴールに向かうパスと次のサブゴールに向かうパスを滑らかに連結できるようなパスを生成しなければならない。ここでは、Figure 4.4 に示すように曲率半径一定のカーブを描いた曲線に沿って移動ロボットを誘導する。

### 4.4.2 シミュレーション

経路上に障害物が存在する場合には、移動ロボットは到達可能なサブゴールを新たに生成し、回避しながら目的とするゴールに向かって移動する。Figure 4.5 は、その様子をシミュレートしたものである。移動ロボットは、回避動作を終えた後、本来のサブゴールを検出してそのサブゴールに向かうことになる。このとき移動ロボットは、サブゴールに向かう途中で障害物に接触している。これは、移動ロボットが視覚センサから得た画像情報だけでは真横の情報を得ることはできず、障害物を通過したことを確認できないからである。このように、障害物の形状、ロボットのまわりの状況によっては、1度よけた側方の障害物に再び接触することがある。視覚センサは、これ

から移動していくさきの情報を大域的に捕らえるのに適しており、真横や真後ろの近接情報は得にくい。したがって、ロボットが障害物の側方を通過したことを確認するセンサなどを用いて近接情報を補う必要がある。第4.5節でセンサ情報処理とセンサの協調について述べる [久保田 87]。

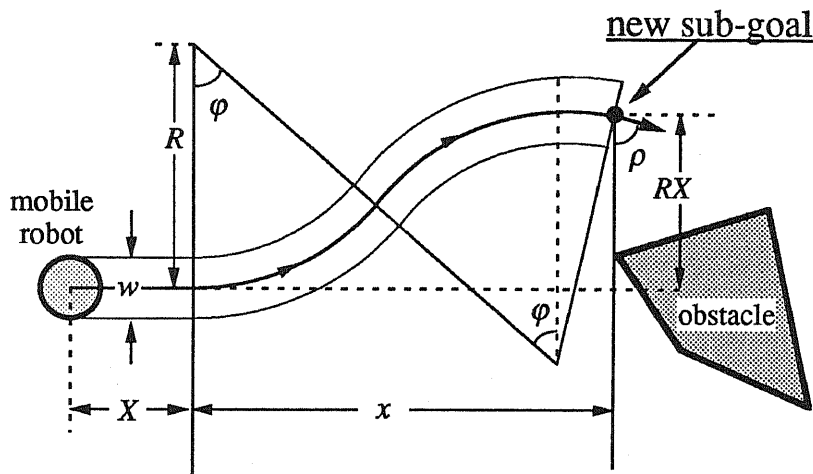


Figure 4.4 Obstacle avoiding path

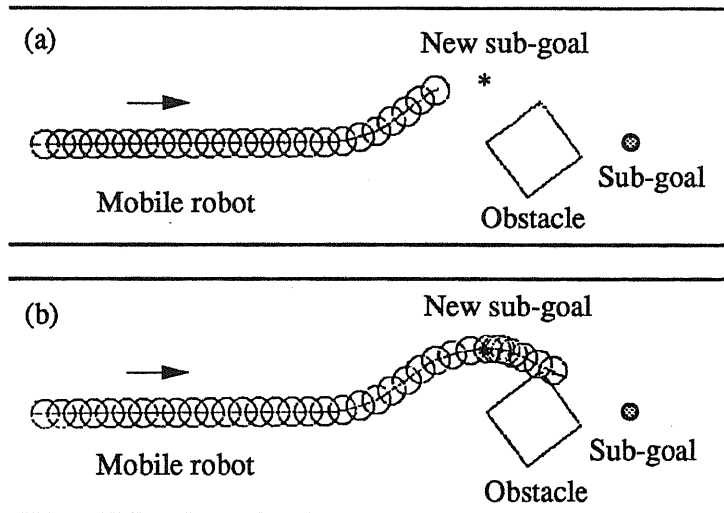


Figure 4.5 Simulation result

## 4.5 センサ情報処理

### 4.5.1 センサ情報のデータ構造

センサから得られる情報はさまざまであり、その処理は場面場面により異なってくる。また種々のセンサから得た情報の中には共通な情報と固有な情報とがある。従ってフレームにまとめることにより処理がしやすくなる [Henderson85][Chiu86]。また、実行レベルでは位置、速度等のデータを用いた数値処理が主であるのに対して、計画レベルではタスクプランニングなど記号処理が主となっている。従ってこの両者の橋渡しを行い、そのレベルで実時間処理が行えるような、より抽象的なデータ構造が必要と思われる [高橋 90]。そこで周囲の環境から得たセンサ情報を次のようなデータ形式で表現することにする。

センサ情報：(対象, 属性, 属性値)

さらに属性値は値、誤差、信頼度、測定時刻からなる。ここで信頼度はセンサデータを融合して同一対象の特性値を求める際に必要とされる尺度であり、センサの信頼性を表す。本システムでは Figure 4.6 に示すようなセンサ情報のデータ構造を用いる。

### 4.5.2 センサ情報の統合・融合の利点

ロボットが現実の世界をとりあつかうためには、必要な情報を信頼性高く獲得することが必要である。一種類のセンサでは正確な情報を獲得することはなかなかできず、また各センサには得意とする情報と苦手とする情報とがある。したがって、各種センサから取得した情報を用いることによって、行動の決定に役立つ信頼性の高い情報を獲得することができる [石川 88][Brady88][Zheng89]。移動ロボットは Table 4.1 に示すような種々のセンサを多重に利用している。移動ロボットは不確実で、未知の環境で作業するため、複雑なタスクを頑健な方法で処理することが要求される。センサデータの統合・融合には、次のような利点が考えられる [Luo89][木下 90][石川 90]。

### <各種センサの誤りの減少>

不正確なセンサデータ解釈モデルによる誤り、センサの故障による誤り、センシングメカニズムに固有な誤りなどがある場合、相補的な性質をもつセンサを多重に用いることにより誤りを減らすことができる。

### <ハイブリッドな情報の生成>

ハイブリッドな情報とは、いくつかのセンサデータの組合せから得られる情報であり、環境内にある既知の物体については新しい物理的な性質を与えたり、また未知の物体に対してはその物体を同定する上で有用となる。

### <センサの性能の向上>

多くのセンサからの情報を統合・融合することにより、センサの補正をしたり、雑音等を除去したりすることができる。

## 4.5.3 センサ情報処理手順

Figure 4.7 にセンサ情報処理の流れ図を示す。各々のセンサは信号の前処理後、センサデータを時々刻々出力している。この出力データはセンサフレームの対応するフレーム内に書き込まれる。センサフレームにあるセンサデータは雑音や誤差を含んでいるので、エラーデータの検出・補正を行ってから利用しなければならない。ロボットの行動決定に必要な情報は信頼性の高いものでなければならず、センサの情報処理においてセンサデータのエラー検出・補正は重要な操作となる。次に、センサ選択ルールに従って必要なセンサ情報を選択し、統合融合処理を行う。このようにして行動決定に役立つ信頼性の高いセンサ情報が得られる。

センサフレームシステム (SFS) には、Table 4.2 に示すような関数があり、このセンサフレームシステムには、次のような利点が考えられる。

1. 種々のセンサから得た情報はすべてセンサフレームにかきこまれるので、個々のセンサにとらわれずにフレームを通してセンサ情報処理を行えばよい。また、センシング処理のモジュール化が可能となる。
2. 各センサフレームの情報を組み合わせることによりハイブリッドな情報が得られる。また、冗長な情報を用いればより精度の高い情報が得られる。
3. センサ情報をフレームの形にまとめることは現在必要としているセンサ情報を少ない手順で効率よく実行するための推論などの手法に向いていると思われる。
4. 関数 `fget`, `fput`, `fremove` はそれぞれ各フレームから値を取り出す操作、フレームに値をいれる操作、フレームの値を削除する操作である。このような関数を用いてデータのアクセス、データの更新、フレームの再編成が容易にできる。

#### 4.5.4 センサデータのエラー検出・補償法

ロボットの行動決定に必要な情報は信頼性の高いものでなければならず、センサの情報処理においてセンサデータのエラー検出・補償は重要な操作となる。例えば超音波センサなどから距離情報を得る場合、その測定精度をあまり高く期待するのは無理である。なぜならば超音波の反射特性は対象物体の表面形状や性質によって大きく左右され、また指向性の問題もあるからである。従ってレンジセンサを多重に利用することにより誤りを減らし、雑音等を除去して、測定値の信頼性と精度を上げなければならない。

ここでは、相関マトリクスを導入した方法について述べる。同じ物体の特性に関する種々のセンサからのデータを融合するためには信頼の尺度が必要である。ここではセンサの特性関数として確率密度関数を用い、エラーデータ判定の基準としては相関マトリクスを導入する [Luo87]。

$k$  番目の物体特性に対するセンサ  $i$ 、センサ  $j$  の確率密度関数を  $P_{ik}(x)$ ,  $P_{jk}(x)$  とする。各確率密度関数は各レンジセンサフレームにおいて時刻  $(t, t + \Delta t)$  の間に得られた距離データより求められる。ここで相関関数  $P_{ijk}$  を次のように表すことにする。

$$P_{ijk} = \int_{X_2}^{X_1} \min\{P_{ik}(x), P_{jk}(x)\} dx \quad (4.17)$$

この  $P_{ijk}$  は Figure 4.8 において斜線部分に相当する。すなわち、 $P_{ijk}$  の値が大きいほどセンサ  $i$  とセンサ  $j$  の  $k$  番目の物体特性に対するデータが密接であることを示している。さらに、相関マトリクス  $P_k$  を次のように定義する。

$$P_k = \begin{bmatrix} P_{11k} & \cdots & \cdots & P_{1mk} \\ \vdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \vdots \\ P_{m1k} & \cdots & \cdots & P_{mmk} \end{bmatrix} \quad (4.18)$$

この相関マトリクスはセンサ同士の関係を表すのに用いられる。いま  $P_k$  を次のような方法でネットワークの形に表すことにする。すべてのセンサをノードとし、ノード間は相関関数の値を添えて結ぶ。このことによってすべてのセンサ間の関係づけをおこなうことができる。強く結びついているセンサ集団は正しいデータを与えていることを示し、弱く結びついているセンサはエラーの疑いがあることを示す。このようにしてエラーデータの検出を行う。この一例を Figure 4.9 に示す。

エラーセンサが検出されたならば、センサのエラーデータの補償を行い、センサフレームのデータを更新する手続きをとる。エラーデータの補償の方法として、Luo らによって Figure 4.10 に示すような方法が提案されている [Luo86]。エラーデータの値  $V_{ij}$  を補償して、融合された値  $V_{ij}^*$  を得ることができる。

#### 4.5.5 センサ情報の統合・融合

移動ロボットが複雑な環境内を移動する際には数種のセンサからの情報による誘導が有効な手段と考えられる。前に述べたように、視覚センサは大域的な情報は得やすいが、正確な近接情報を得るのが難しい。そこで視覚センサとレンジセンサとを協調させることにより、近接情報を融合してより正確なコントロールを行うための情報をつくるのが望ましい。視覚センサ情報は大域的な情報であり、障害物の検出、回避ポイントの生成、サブゴールへの追従、移動経路の生成、環境の予測情報に使われる。レンジセンサ情報はロボットのまわりの近接距離情報であり、周囲の環境認識、予測された環境の確認、移動ロボットの位置同定に使われる。視覚センサ情報とレンジセンサ情報の協調的処理方法としては次の3つが考えられる。

1. レンジセンサにより、前方の障害物の検出（障害物の有無・障害物までの距離）を行い、視覚センサにより障害物を確認し、特徴を抽出する。障害物までの距離をあらかじめレンジセンサにより得られるため、視覚センサの画像処理の短縮化ができる。
2. 視覚センサにより得た画像データはこれから進んで行く場所のデータなので予測情報という性質をもつ。したがって、サブゴール付近でどのセンサを選択したらよいかという判断に用いることができる。また、レンジセンサによりサブゴール到達の確認ができる。
3. 移動ロボットは画像のサンプリングごとにその時点におけるサブゴールの位置を検出する。そして、そのサブゴールの方向を示す視覚センサベクトルを得る。一方、レンジセンサからの情報によって障害物の位置を表すレンジセンサベクトルを得る。得られたベクトルを合成することによって、観測と行動の有機的な結合を行うことができる。

```
(vision
  (object_1
    (distance (value(v11), error(e11), conf(c11), time(t11)))
    (orientation (value(v12), error(e12), conf(c12), time(t12)))
    (size (value(v13), error(e13), conf(c13), time(t13))))))
(range_sensor_A
  (object_2
    (distance (value(v11), error(e11), conf(c11), time(t11)))
    (orientation (value(v12), error(e12), conf(c12), time(t12))))))
```

Figure 4.6 Sensor data structure

Table 4.1 Robot sensor

Terminal sensor	Touch sensor Contact force sensor Slip angle sensor Proximity sensor Working sensor
Force sensor	Servo deflection sensor Joint torque sensor Wrist sensor
External sensor	Visual sensor Acoustic sensor Collision sensor Temperature and humidity sensor
Position detection sensor	Guide sensor Land mark sensor Visual sensor Ultra-sonic sensor Mobility detection sensor
Safety sensor	Obstacle detection sensor Abnormality detection sensor



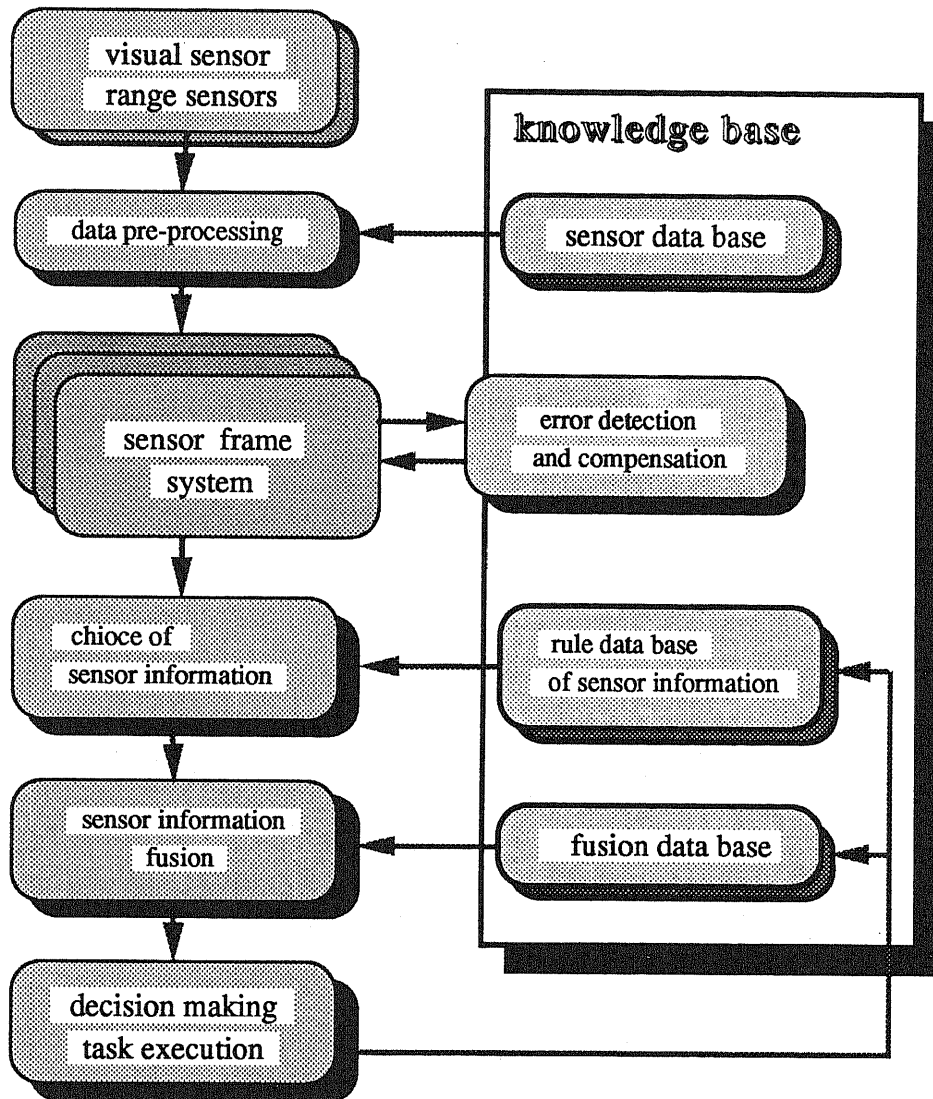


Figure 4.7 Procedure of sensor information processing

Table 4.2 Basic functions of SFS

FGET	: to get information from SFS
FPUT	: to put information into SFS
FREMOVE	: to omit information from SFS
FUPDATE	: to update information in SFS

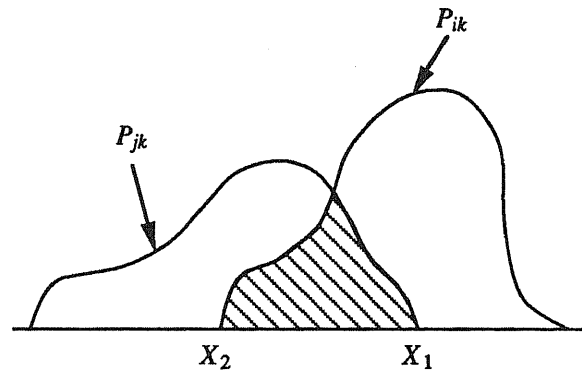


Figure 4.8 Probability density function

$$P_k = \begin{bmatrix} 1.0 & 0.9 & 0.7 & 0.5 & 0.1 \\ 0.9 & 1.0 & 0.8 & 0.4 & 0.3 \\ 0.7 & 0.8 & 1.0 & 0.7 & 0.0 \\ 0.5 & 0.4 & 0.7 & 1.0 & 0.1 \\ 0.1 & 0.3 & 0.0 & 0.1 & 1.0 \end{bmatrix}$$

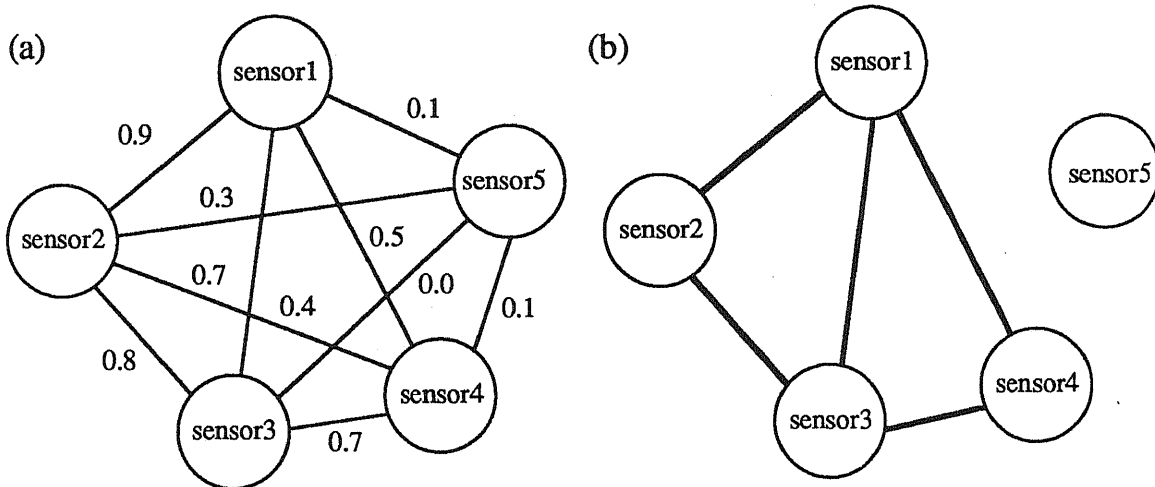


Figure 4.9 Example of sensor error detection

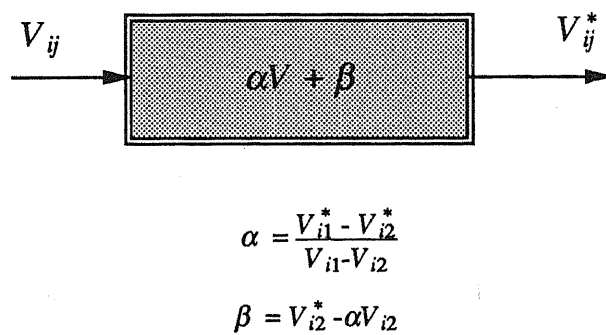


Figure 4.10 Error data compensation

## 4.6 視覚センサとレンジセンサとの協調

### 4.6.1 移動ロボットにおけるセンサフュージョン

一般に、単一のセンサでは得られる情報量が少ないため、認識できる環境には限度がある。移動ロボットが複雑な環境内を移動する際には数種のセンサからの情報による誘導が有効な手段と考えられる [Shafer86]。

前に述べたように、視覚センサは大域的な情報は得やすいが、正確な近接情報を得るのが難しい。これを解決するために視覚センサと超音波センサなどのレンジセンサを協調させることにより近接情報を融合して、より正確なコントロールを行うための情報をつくるのが望ましい。視覚センサ情報は大域的な情報であり、障害物の検出、サブゴールの生成・追従、移動経路の生成、環境の予測情報に使われる。レンジセンサ情報はロボットのまわりの近接距離情報であり、周囲の環境認識、予測された環境の確認、移動ロボットの位置同定に使われる。視覚センサ情報とレンジセンサ情報処理手法として次の3つの方法を考える [Kubota88]。

#### 1. 複合的処理

視覚センサによりこれから進んで行く場所の環境情報を獲得する。一方、レンジセンサによりロボットのまわりの環境情報を得て側方および後方の障害物を検出できる。これより、局所性の回避および測定レンジの拡大が可能となる。(Figure 4.11 参照)

#### 2. 統合的処理

レンジセンサにより、前方の障害物の検出(障害物の有無・障害物までの距離)を行い、視覚センサにより障害物を確認し、特徴を抽出する。障害物までの距離をあらかじめレンジセンサにより得られるため、視覚センサの画像処理の短縮化が可能となる。また、視覚センサによって測定した障害物までの距離情報やレンジセンサの測定値には、測定誤差が含まれている。例えば、カルマンフィルタを用いることによって障害物の位置情報の推定を行い、精度を向上させることができる。(Figure 4.12 参照)

### 3. 融合的処理

移動ロボットは画像のサンプリングごとにその時点におけるサブゴールの位置を検出する。そして、そのサブゴールの方向を示す視覚センサベクトルを得る。一方、レンジセンサからの情報によって障害物の位置を表すレンジセンサベクトルを得る。得られたベクトルを合成することによって、移動ロボットの行動決定を行うことができる。このようにして、観測と行動の有機的な結合が実現できる。(Figure 4.13 参照)

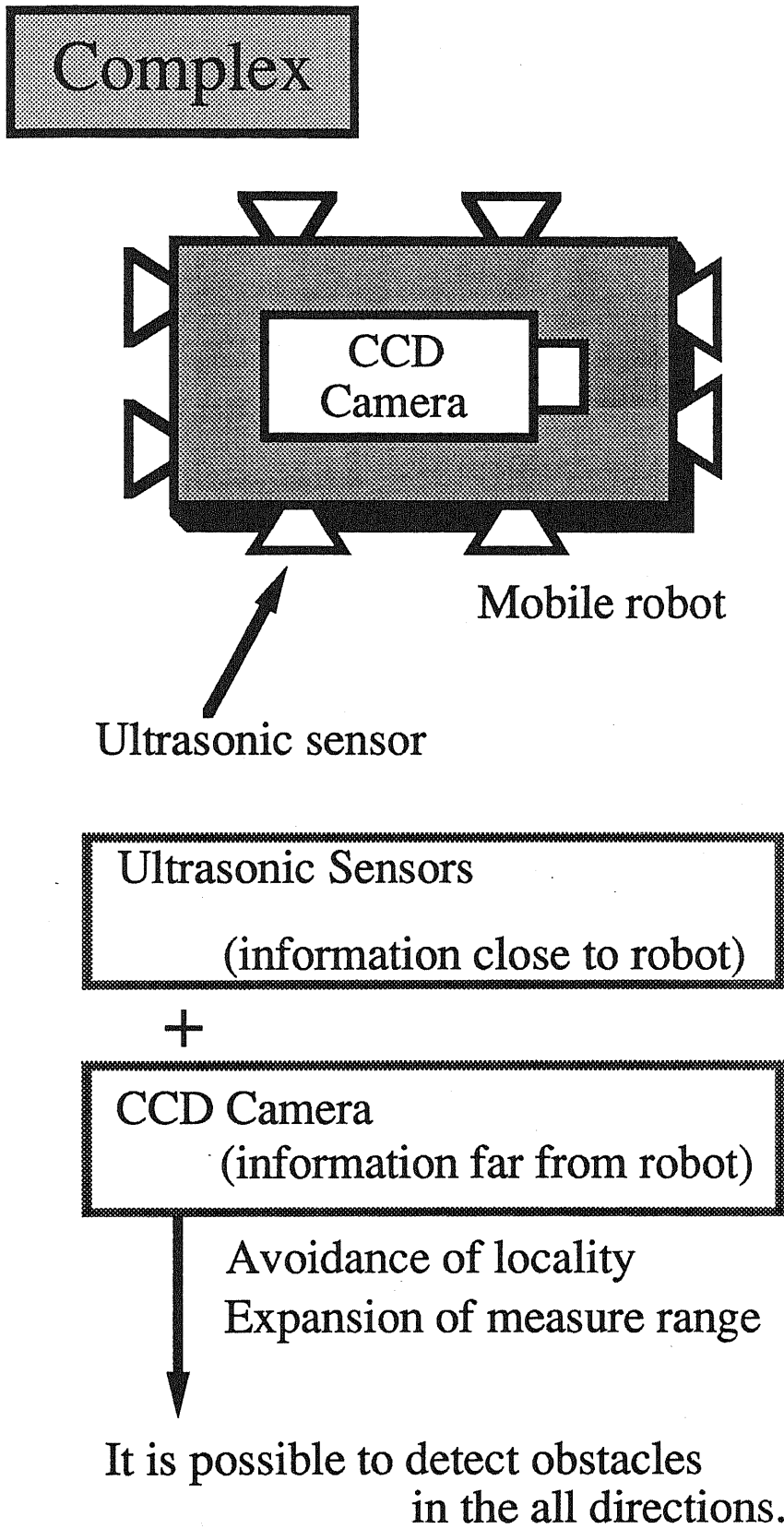


Figure 4.11 Sensor complex method

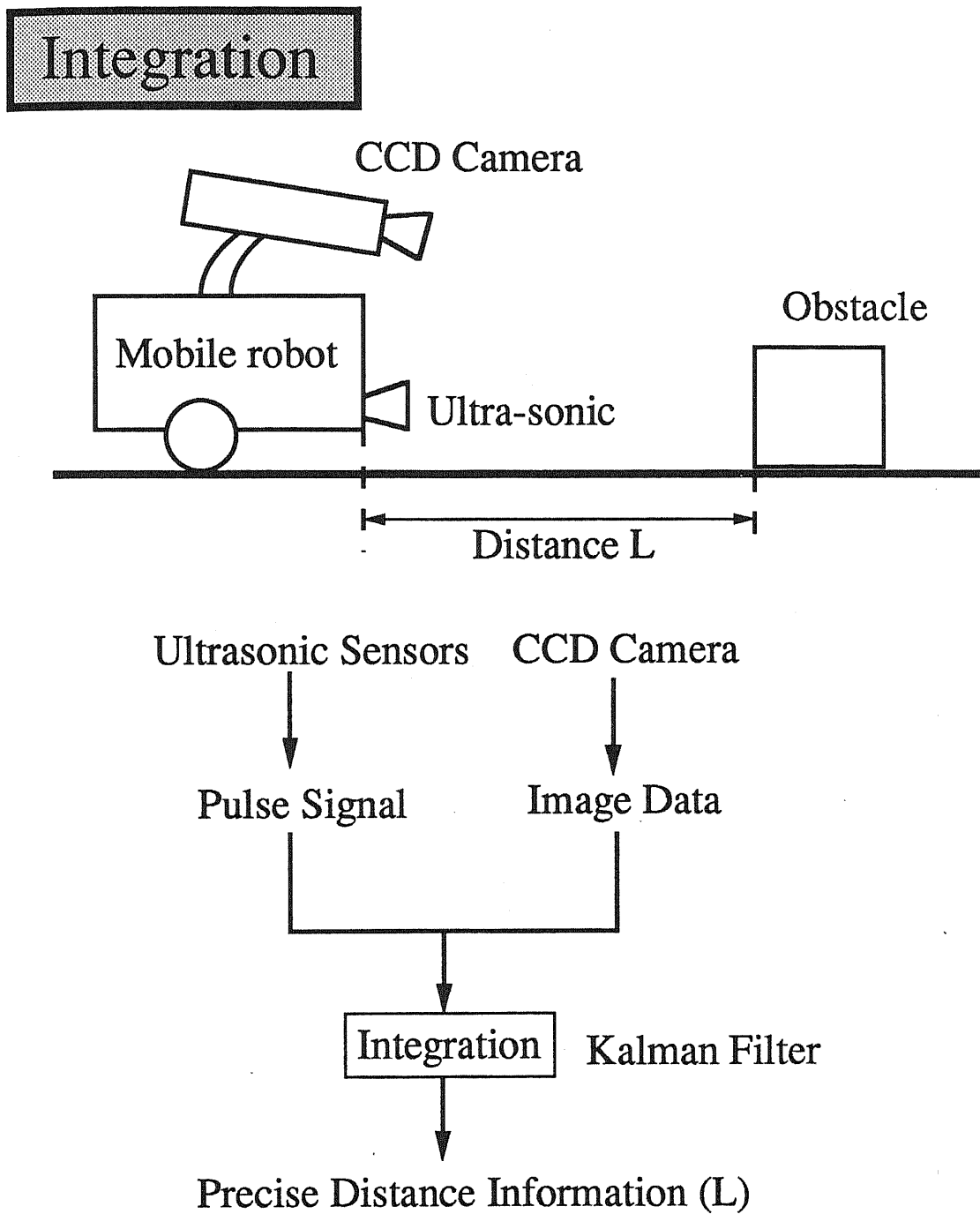


Figure 4.12 Sensor integration method

# Fusion

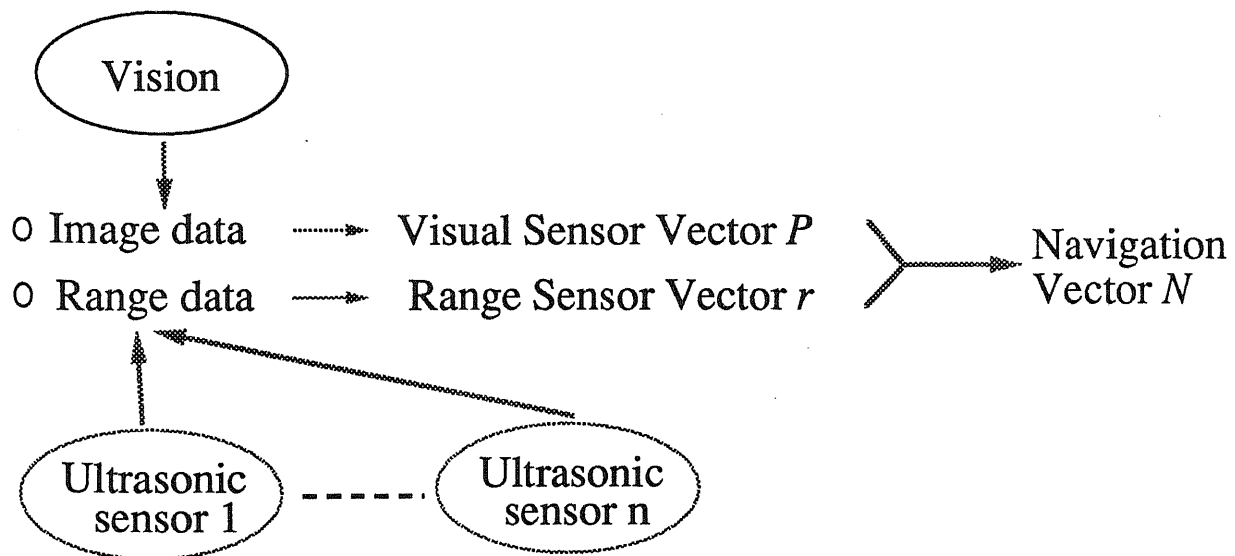
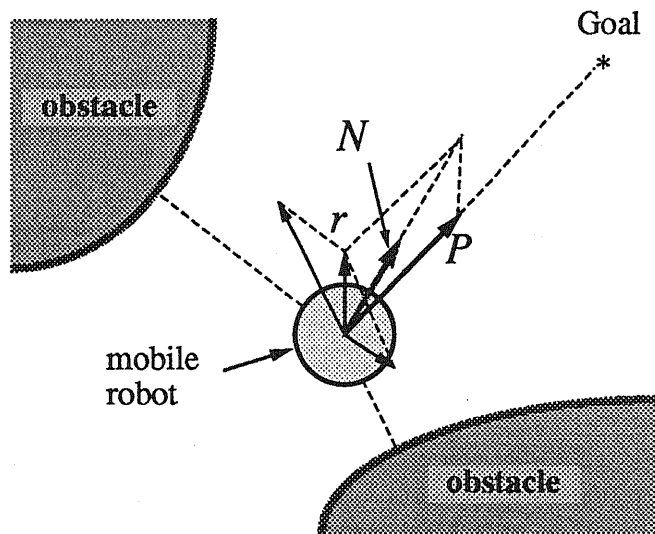
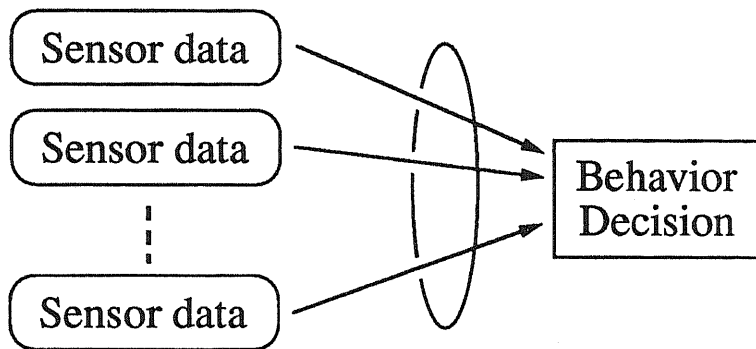


Figure 4.13 Sensor fusion method

## 4.6.2 カルマンフィルタによるセンサ統合

ある物理量を測定する場合、その物理量の実現値それ自身が直接取り出せることはまれであり、1つの測定行為がいくつかの複雑なプロセスからなっていることが多い。測定値には、測定の際の外乱、雑音、測定誤差が含まれている。その測定値からいかによりよい推定値を導くかが問題となる。

ここでは、障害物の位置をカルマンフィルタを用いて推定することを試みる[有本][片山]。いま、移動ロボットからみた障害物までの距離  $x(t)$  のみを考える。障害物の位置を  $x(t)$ 、相対速度を  $\dot{x}(t)$ 、相対加速度を  $\ddot{x}(t)$  とし、視覚センサおよびレンジセンサによってその位置がサンプル周期  $\Delta$  ごとに測定できるものとする。そこで、

$$x_k = \begin{bmatrix} x(k\Delta) \\ \dot{x}(k\Delta) \\ \ddot{x}(k\Delta) \end{bmatrix} \quad (4.19)$$

$$y_k = \begin{bmatrix} y_{ak} \\ y_{bk} \end{bmatrix} \quad (4.20)$$

ただし、

$y_{ak}$  : 視覚センサからの測定値  
 $y_{bk}$  : レンジセンサからの測定値

とすると、このシステムは次のように表される。

$$\begin{cases} x_{k+1} = Ax_k + Bu_k \\ y_k = Cx_k + w_k \end{cases} \quad (4.21)$$

ただし、

$$A = \begin{bmatrix} 1 & \Delta & \frac{1}{2}\Delta^2 \\ 0 & 1 & \Delta \\ 0 & 0 & 1 \end{bmatrix} \quad (4.22)$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (4.23)$$



$$C = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad (4.24)$$

$$w_k = \begin{bmatrix} w_{ak} \\ w_{bk} \end{bmatrix} \quad (\text{測定誤差}) \quad (4.25)$$

$u_k$ : 時刻  $k\Delta$  から時刻  $(k+1)\Delta$  までの加速度の変動

ここで、 $u_k$  と  $w_k$  を白色雑音とみなし、システム (4.21) にカルマンフィルタの計算法を適用すると次のようになる。

いま、 $x_0$ 、 $u_k$ 、 $w_k$  をガウス性とし、

$$Eu_k = \bar{u}_k \quad (4.26)$$

$$E(u_k - \bar{u}_k)(u_l - \bar{u}_l)' = \delta_{kl}U_k \quad (4.27)$$

$$Ew_k = \bar{w}_k \quad (4.28)$$

$$E(w_k - \bar{w}_k)(w_l - \bar{w}_l)' = \delta_{kl}W_k \quad (4.29)$$

と仮定すると、 $x_k$  の最尤推定値  $\hat{x}_k$  は次のようになる。

$$\hat{x}_k = \tilde{x}_k + P_k C_k W_k^{-1} \{y_k - (C_k \tilde{x}_k + \bar{w}_k)\} \quad (4.30)$$

ただし、

$$\tilde{x}_k = A\hat{x}_{k-1} + B\bar{u}_{k-1} \quad (4.31)$$

$$P_k = (M_k^{-1} + C_k' W_k^{-1} C_k)^{-1} \quad (4.32)$$

$$M_k = A_{k-1} P_{k-1} A_{k-1}' + B_{k-1} U_{k-1} B_{k-1}' \quad (4.33)$$

カルマンフィルタを用いたシミュレーション結果を Figure 4.14 に示す。このように、カルマンフィルタで推定計算をおこない、多数回測定することにより精度を向上することができる。

## 4.6.3 センサベクトル合成法によるセンサ融合

ここでは1つの視覚センサとまわりにレンジセンサを有する移動ロボットについて考える [Kubota87]。視覚センサ情報とレンジセンサ情報との融合方法を Figure 4.15 に示す。移動ロボットは画像のサンプリングごとにその時点におけるサブゴールの位置を検出する。そしてそのサブゴールの方向を示すベクトルとして視覚ベクトル  $p(n)$  を得る。

$$p(n) = \frac{b(n) \cdot (s - m)}{|s - m|} \quad (4.34)$$

ただし、

- $s$  : サブゴールの位置ベクトル
- $m$  : 移動ロボットの位置ベクトル
- $b$  : 係数
- $n$  : サンプリング回数

レンジセンサベクトル  $r(n)$  はポテンシャルエネルギーの概念を用いて次のようになる。

$$r(n) = \sum_i \hat{r}_i(n) \quad (4.35)$$

$$\hat{r}_i(n) = -\frac{w_i \cdot r_i(n)}{|r_i(n)|^3} \quad (4.36)$$

式(4.34)において移動ロボットからサブゴールへのベクトル  $(s - m)$  は視覚センサフレームにおけるサブゴールに対する距離と方向により決定される。例えば次のように求められる。

$$|s - m| = \text{FGET}(\text{visualsensors}, \text{subgoal}, \text{distance}) \quad (4.37)$$

式(4.37)において、visualsensors は Figure 4.6 のセンサフレーム名、subgoal は対象に相当する。 $b(n)$  は  $p(n)$  の大きさを決めるもので、例えばロボットが現在向かっているサブゴールに停止するような場合は  $b(n)$  の大きさを次第に小さくしていけばよい。式(4.36)における  $r_i(n)$  はセンサ  $i$  によって検出される障害物の位置を表すベクトル

で、 $w_i$  は重みを表し、近くに存在する障害物にたいしては大きい値を、遠くにある物体にたいしては小さい値を与えるように決定する。例えば、Figure 4.16 のように与えることができる。式(4.36) で表されるレンジセンサベクトル  $\hat{r}(n)$  は一種のポテンシャルを形成し、障害物との衝突危険度を表している。

視覚ベクトル  $p(n)$  及びレンジセンサベクトル  $r(n)$  を合成することにより、ロボットが次に移動する方向が決まる。時刻  $(n+1)$  における速度  $v(n+1)$  及び、移動方向の修正ベクトル  $a(n)$  は次のようになる

$$v(n+1) = \frac{p(n) + r(n)}{|p(n) + r(n)|} |v(n)| \quad (4.38)$$

$$a(n) = p(n) + r(n) - v(n) \quad (4.39)$$

この方法は超音波センサを中心にした移動ロボットにおけるポテンシャル航法と異なり、視覚でサブゴールを検出しながら移動を行っているため、パスは必ず生成される。レンジセンサと組合せた場合、デッドロックの可能性も生じるが、式(4.34)の  $b(n)$  を大きくして、視覚によるセンシングを大きくすることによってデッドロック状態を回避することが可能になる。このように視覚センサ情報とレンジセンサ情報をうまく融合することにより、障害物に衝突することなく移動ロボットを誘導することができる。

#### 4.6.4 シミュレーション結果

視覚センサとレンジセンサの協調による、ナビゲーションのシミュレーション結果を Figure 4.17 に示す。移動ロボットは視覚センサによって新たなサブゴールを生成し、それに向かって進んでいく。Figure 4.17(b) は、移動ロボットが視覚センサのみを用いて移動した場合で、ロボットのまわりの近接情報が得られていないため、次のサブゴールに向かっていく途中で障害物に接触している。Figure 4.17(c) は視覚センサとレンジセンサの協調をおこなった場合で、移動ロボットは障害物に会うたびに新しいサブゴールを生成し、障害物回避を行っている。ここでは、レンジセンサから近接情報を得ているため、障害物に衝突することなく与えられたサブゴールに到達している。

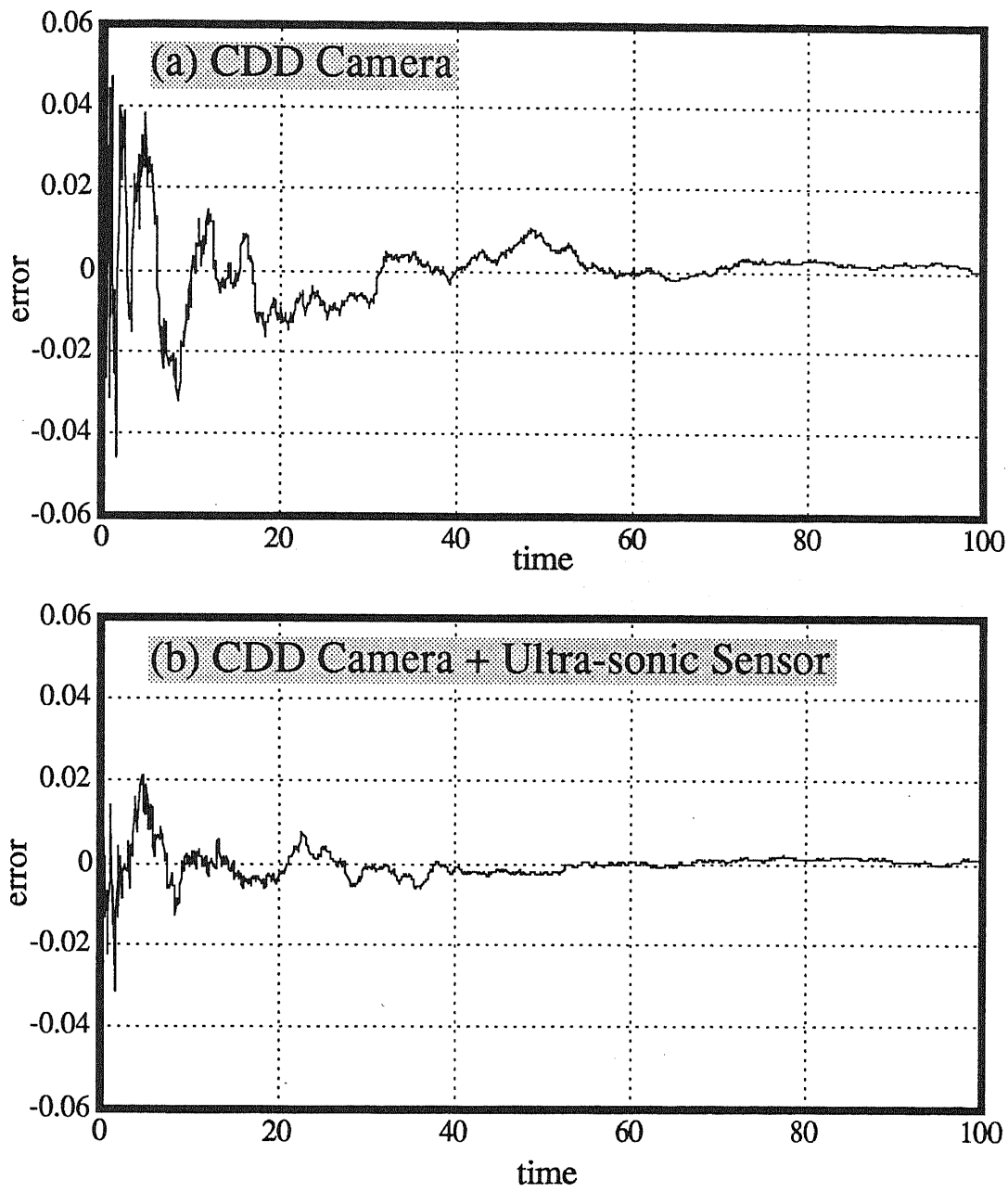


Figure 4.14 Simulation results

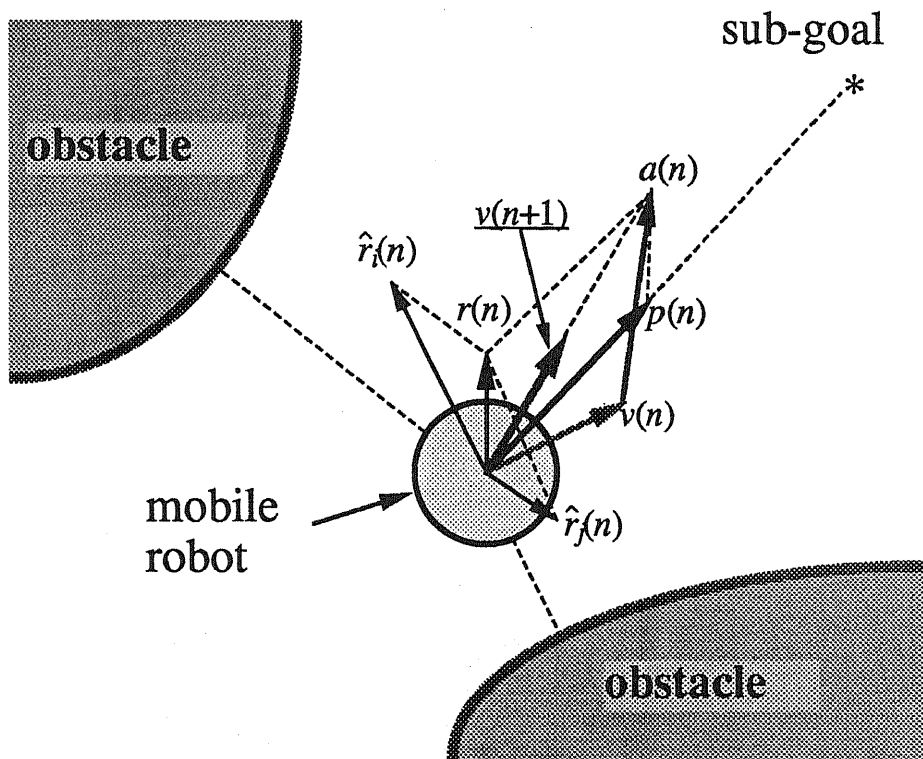


Figure 4.15 Sensor information fusion

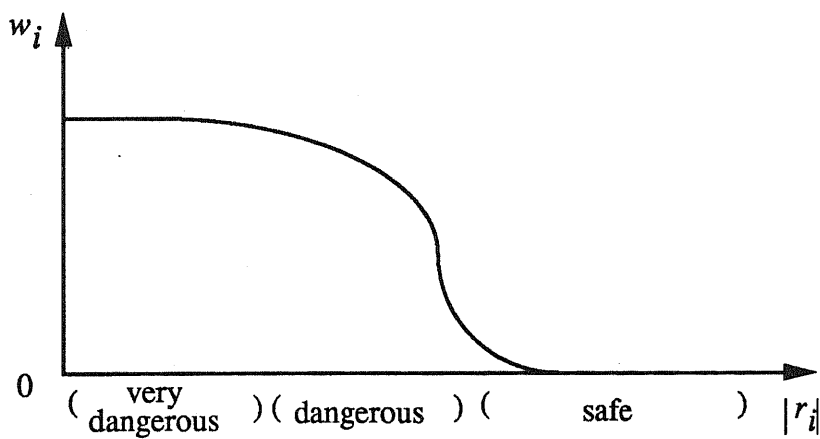


Figure 4.16 Value of  $w_i$

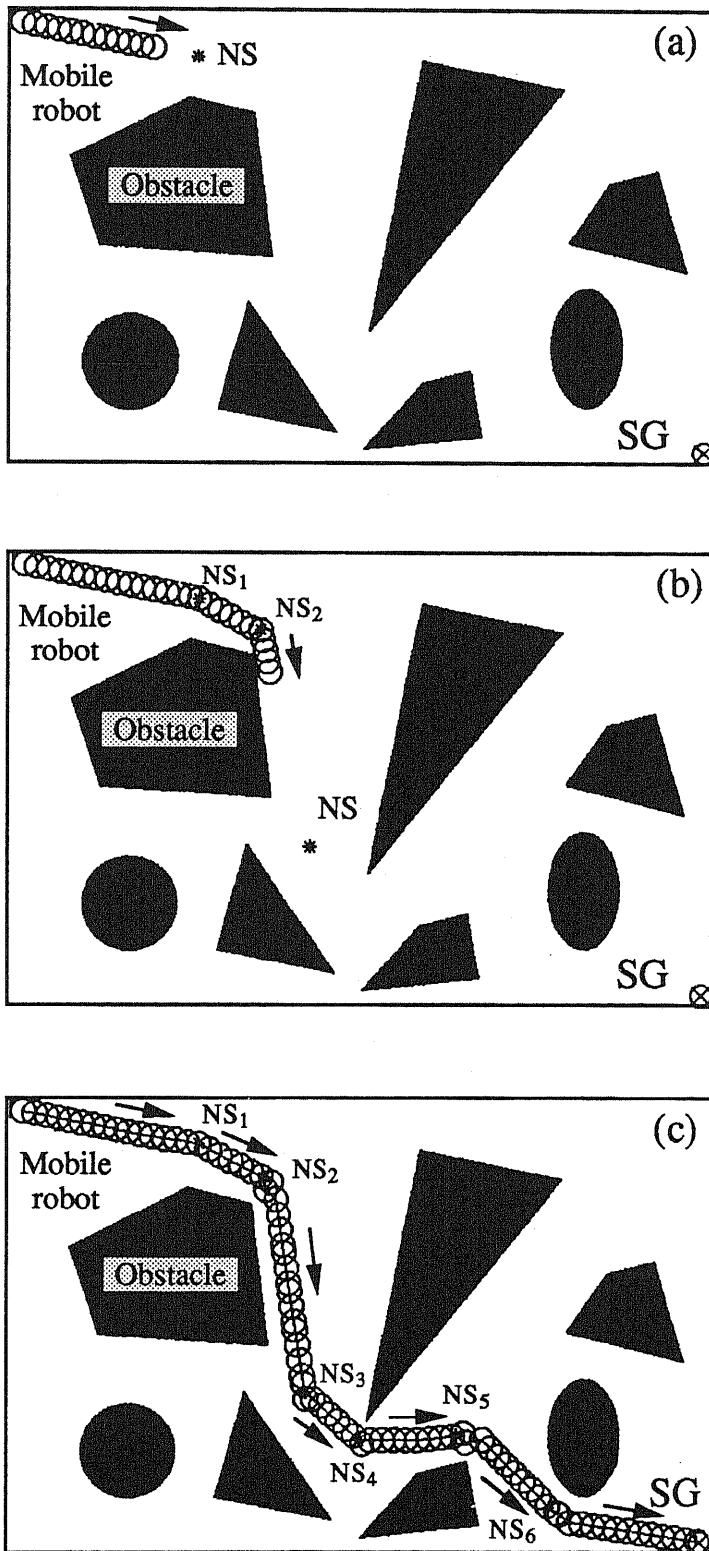


Figure 4.17 Simulation result

## 4.7 環境に応じた行動パターンの生成の試み

移動ロボットやマニピュレータの障害物回避における知識の利用、学習に関していくつかの試みがあるが[比留川 85][村松 87][鈴木 87]、学習の結果として獲得された知識が、別の環境でどのくらい有効に適用されるかが問題となる。この有効性を検証することは、非常に困難と思われるが、比較的単純な知識表現を考えることによりその有効性の予測を行うことができると思われる。

視覚センサ情報とレンジセンサ情報の融合をおこなったが、そのときの重み  $w$  はあらかじめ経験的にきめておいた。しかし、この  $w$  の値は状況に応じて決める必要がある。そこで、移動ロボットが移動を繰り返し行いながら得られたデータを基にして  $w$  の値を更新していく、一種の学習を試みる。

### 4.7.1 知識の表現

知識は、Figure 4.18 に示すようにプロダクションシステムによって表現する。プロダクションルールの condition 部分には、移動ロボットの周囲の状況を与え、action 部分には、移動ロボットが動く方向を与える。移動ロボットの周囲の状況は、目標までの位置を表す視覚ベクトル  $p$  と障害物の位置を表すレンジセンサベクトル  $r$  によって表される。いま、ある時刻  $n$  での状況を  $C_n$  とするとつぎのように表される。

$$C_n : [p, r_i] \quad (i = 1, 2, \dots) \quad (4.40)$$

また、 $C_n$  において移動すべき方向を  $A_n$  とするとつぎのようになる。

$$A_n : \frac{p(n)}{|p(n)|} + \sum w_i \frac{r_i(n)}{|r_i|^3} \quad (4.41)$$

したがって、時刻  $n$  での状況  $C_n$  で  $A_n$  が適用されることは、

$$(C_n, A_n) \quad (4.42)$$

と表現される。

### 4.7.2 学習

1回の試行で得られた評価を時系列データとして扱い、ここから新しい知識を得て  $w$  の値の更新を行う。

いま、 $C_n$  で  $A_n$  を適用した時の評価を  $E_n$  とする。試行が1回行われるとつぎのようなデータが記録される。

$$(C_j, A_j, E_j) \quad (j = 1, \dots, m) \quad m: \text{ステップ数} \quad (4.43)$$

$E_i$  はある基準を基にして定義された評価関数  $E$  の値として与えられる。ここで与えた評価の列は、すべて互いに関係しあっていると考えることができ、この評価の列を時系列データとみなしてグラフにすることができる。

評価関数としては移動における特徴量を正確にとらえたものがよいが、ここではつぎのような評価関数  $E$  を用いる。

$$E: \frac{|p(n)| - |p(n-1)|}{|p(n)|} - \sum \frac{|r_i(n)| - |r_i(n-1)|}{|r_i|} \quad (4.44)$$

$E$  の値は目標点に近づけば小さくなり、障害物に近づけば大きくなる。したがって、この評価関数をもとにグラフをつくった場合、グラフは0を基準として負のとき良いふるまい、正のとき悪いふるまいとなる。

$C_n$  において  $A_n$  の適用が成功ならば評価値は0より小さな値になるので、スタートからゴールまですべて成功した場合、グラフは負の領域で高い値を保持することになる。悪いふるまいをしている点は極大点に現れ、あるしきい値より大きな点を注目点としてその極大の部分なくなるように、重み  $w$  の値を更新する知識を生成する。



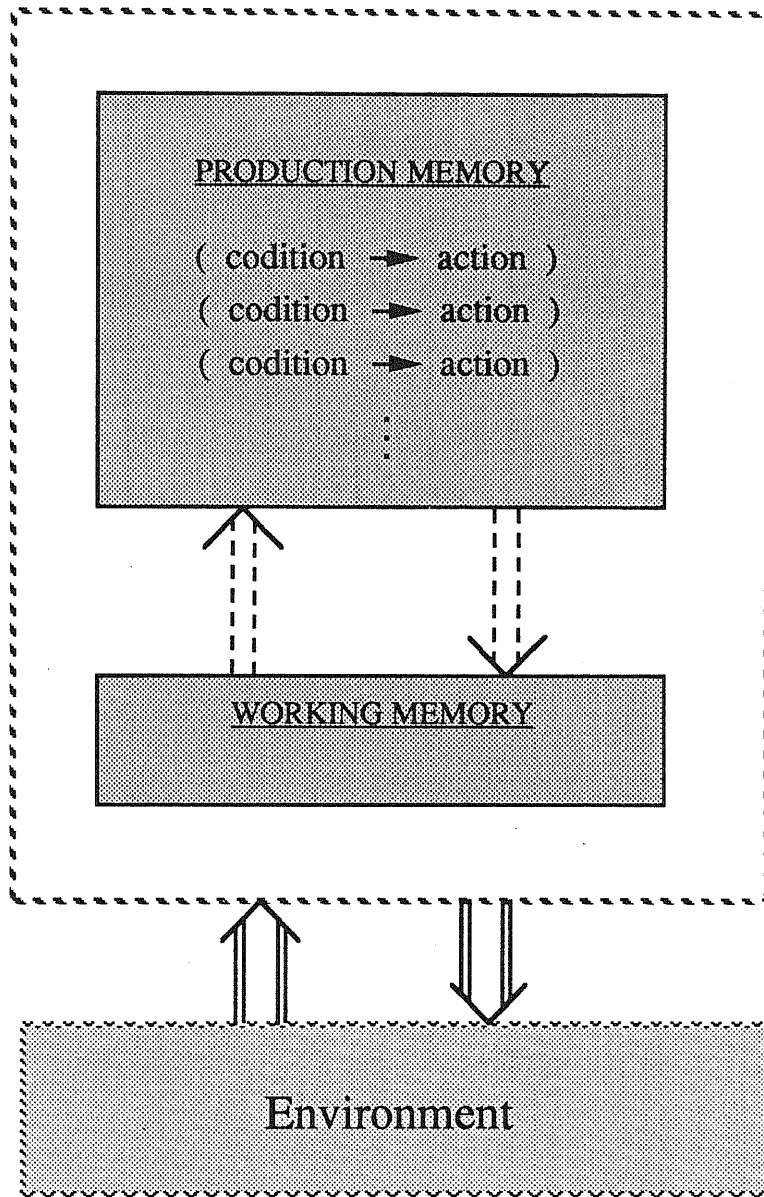


Figure 4.18 Production system

### 4.7.3 シミュレーション結果

シミュレーション結果を Figure 4.19、Figure 4.20 に示す。Figure 4.19(a) は学習前の移動ロボットの移動の様子をシミュレートしたもので、 $w$  の値としては Figure 4.19(b) に示すものを初期値として与えた。そのときの評価関数の値を Figure 4.19(c) に示す。5 回試行をおこなったあとの実行結果を Figure 4.20 に示す。移動ロボットは学習前よりもなめらかに障害物を回避してゴールに到達している。

しかしながら、評価グラフを見る限りにおいて、効率のよい学習とはいえないようである。また、学習により得られた知識の爆発的な増加に対する対策、学習による重み  $w$  の収束性などについて検討を行う必要がある。さらに、極大点の中から注目点と決めるしきい値が大きすぎると局所的な知識が増大し、小さすぎると十分な学習効果が得られない。したがって、しきい値をどのように決めるか、またメタレベルの知識についても検討する必要がある。

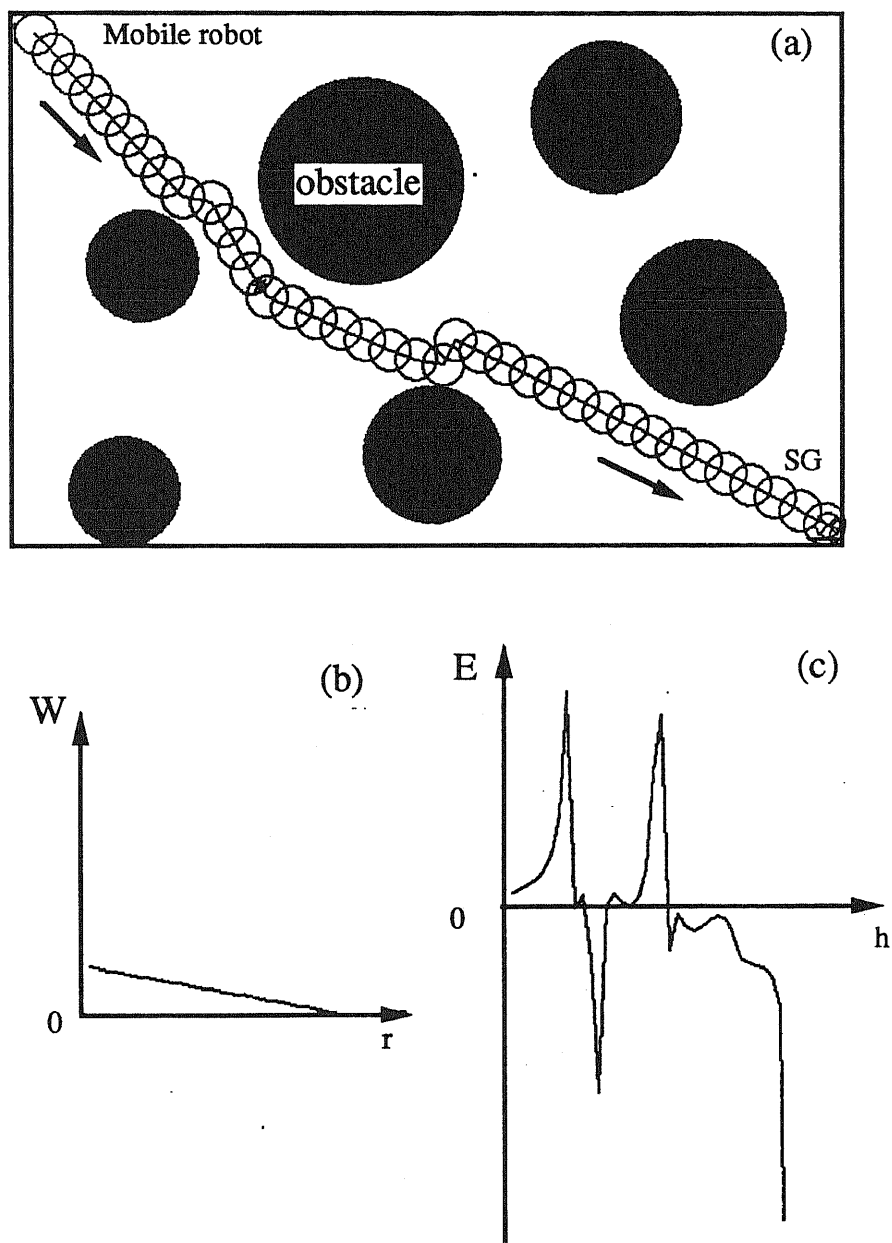


Figure 4.19 Simulation result (before learning)

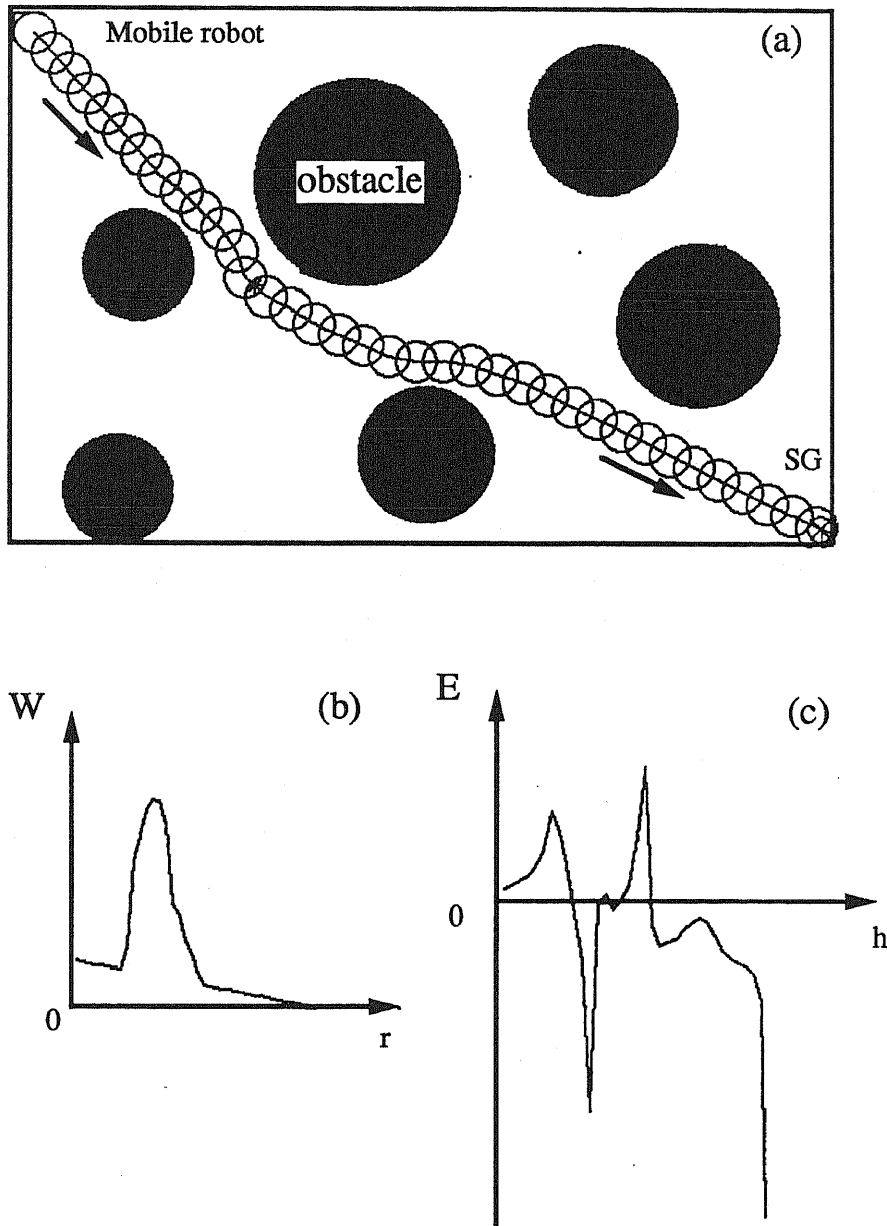


Figure 4.20 Simulation result (after learning)

## 4.8 移動障害物に関する考察

ロボット技術の進歩によって、複数台の移動ロボットが同一環境内で同時に移動するような状況、あるいは人間と共存して移動する状況は十分に考えられる。このような場合にはロボットとロボット、あるいはロボットと人間との衝突回避の問題を扱う必要が生じる。障害物が移動する場合の簡単なシミュレーションを行ってみた。そのようすを Figure 4.21 に示す。これは障害物が非常にゆっくり動くものとし、視覚センサとレンジセンサの両方を用いて衝突回避を行っている。視覚センサによって画像のサンプリングごとに回避ポイントの更新を行い、また回避ポイント近くではレンジセンサから近接情報を取り入れており、移動障害物に衝突することなく、ゴールに到達している。

しかしながら、実際にはもっと複雑な環境を想定しなければならない。したがって、移動障害物に対して、それが何であるか識別することよりも、一刻も早くその物体の移動速度や進行方向を検出し、それに対する適切な行動計画をたてる必要がある。これに関しては次章で述べる。

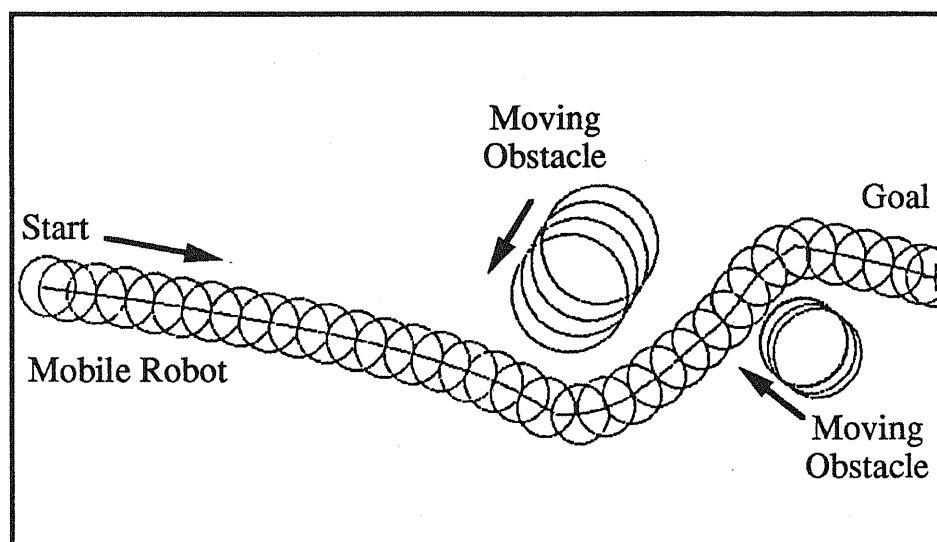


Figure 4.21 Simulation result for moving obstacles

## 4.9 実験による検討

### 4.9.1 実験用移動ロボットシステム

実験に用いた移動ロボットは、主な構成要素としてCCDカメラを搭載した移動ロボット本体と、画像処理、プランニング、ナビゲーション、コントロール等を行うホストコンピュータ、および超音波センサからの信号処理を行うマイクロコンピュータからなる。移動ロボットシステムの外観図および構成を Figure 4.22 および Figure 4.23 にそれぞれ示す。

ロボットには、一对のステッピングモータ駆動輪および一对のボールキャスターが取り付けられており、前後直進とその場回転とができる。駆動輪の制御は、ロボットに搭載された、Z80ボードとステッピングモータ駆動用のドライブ回路を介し、ホストコンピュータから制御パルスを送信することにより行う。また、ロボットの頭部には小型のCCDカメラ (SONY XC38) 1台が取り付けられており、移動しながら画像を取り込むことができる。CCDカメラによって取り込まれた画像情報は、ホストコンピュータに取り付けられたイメージメモリボード (ALTEC ALT-256-8) を通じて、解像度  $256 \times 256$ 、輝度レベル8bitの濃淡画像データに変換され、ホストコンピュータのVRAM上へ一画像当り、約0.03sec以内にDMA転送される。

さらに、移動ロボットには、下部の前後左右の4方向に超音波センサ (RS-2410) が取り付けられており、ローカルな距離情報を取り込むことができる。超音波センサ (RS-2410) は、ロボットの前後左右の4方向にトランスデューサ (T-3C) とレシーバ (R-3C) の1組ずつが取り付けられている。超音波としては、40kHz (8.5mm) を用いており、0.6msec 間発信、50msec 間受信という動作を毎秒20回繰り返す。受信された反射波は、先着優先回路を経て、最も近い障害物までの距離を反射時間に対応するパルスとして出力する。このパルスをデジタルカウンタによって距離値としてカウントし、3桁BCD符号 (12bit) として出力する。このデジタルカウンタには、十進3桁の表示用の7 segment 出力のカウンタとBCD出力のカウンタの2種類のカウンタがあり、全方向合わせて計8つのカウンタが搭載されている。3桁BCD出力は (4方向合わせて48bit) は、超音波センサデータ処理用のマイクロコンピュータに送られ、所定のデータ処理を行う。超音波センサのシステム構成を Figure 4.24 に示す。また、各信号線の波形およびタイミングを Figure 4.25 に示す。

超音波センサのサンプリングタイムは、画像データの処理やプランニングに比べて、かなり短いので独立のマイクロコンピュータを用いて、リアルタイムに更新する。ホストコンピュータとは、24bit のパラレル通信によって結ばれており、ホストコンピュータが必要に応じて超音波センサからのデータをアクセスできるようになっている。

#### 4.9.2 目標追従実験

まず、視覚情報をフィードバックすることにより移動ロボットを目標に追従させる実験を行った。目標としては、画像処理に負担をかけないために背景とのコントラストのつよい円柱を用い、目標の特徴（円柱の高さや幅など）はあらかじめロボットに与えられている。実験において移動ロボットの移動速度は  $12.8[\text{cm}/\text{sec}]$ 、サンプリングタイムは  $0.685[\text{sec}]$ 、障害物までの距離は  $2.0[\text{m}]$  であった。移動ロボットに搭載された CCD カメラの視野が狭いこと、および実験環境の制約から目標までの距離を大きくできなかつたことから、ゲインをあまりあげることができなかつたが、移動ロボットは画像データをフィードバックしながら目標に追従した。

#### 4.9.3 センサ協調実験

ここでは、ゴール（あるいは与えられたサブゴールと考えてもよい）の位置を既知とし、また実験装置の制約から、障害物が1個存在する場合について実験を行った。実験に用いた移動ロボットのシステムの外観図及び実験において右側の超音波センサから得られたレンジセンサ情報をそれぞれ Figure 4.26、Figure 4.27 に示す。また、実験全体のおおよそのようすを Figure 4.28 に示す。移動ロボットは、超音波センサによって近接情報を得ているので、障害物に衝突することなく、ゴールに到達することができた。このとき、移動ロボットの速度は約  $13[\text{cm}/\text{s}]$  で、画像処理時間は約  $0.6[\text{s}]$  であった。

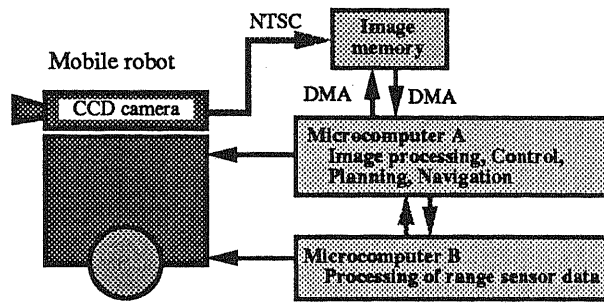


Figure 4.22 Experimental mobile robot system

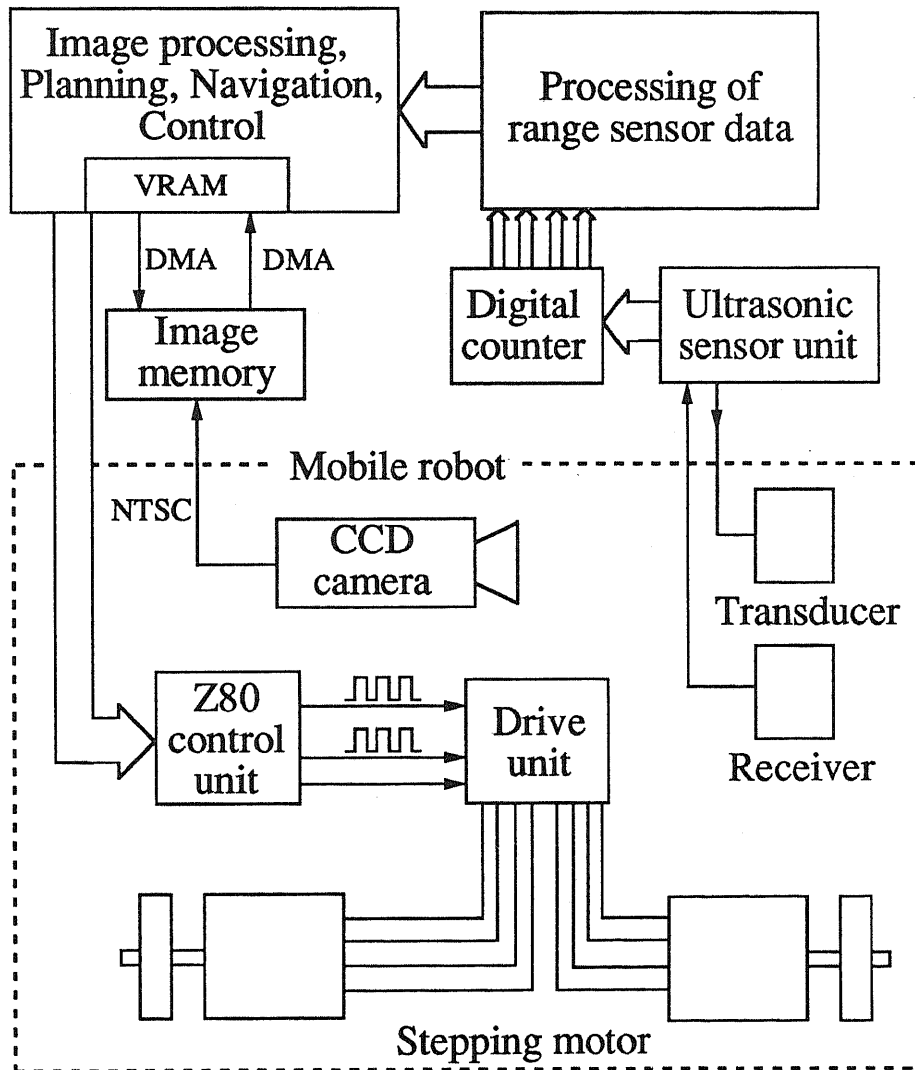


Figure 4.23 System configuration



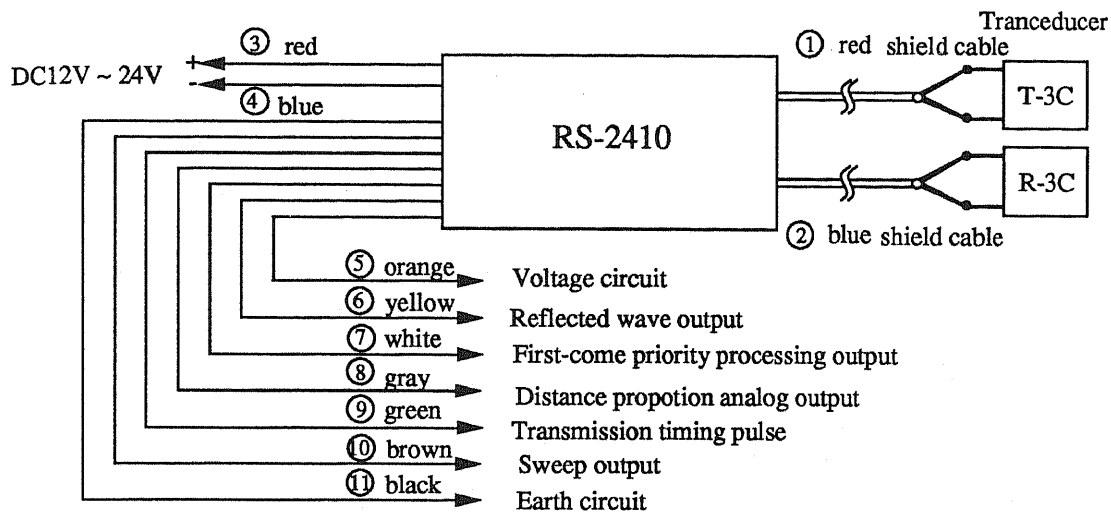


Figure 4.24 Ultra-sonic sensor

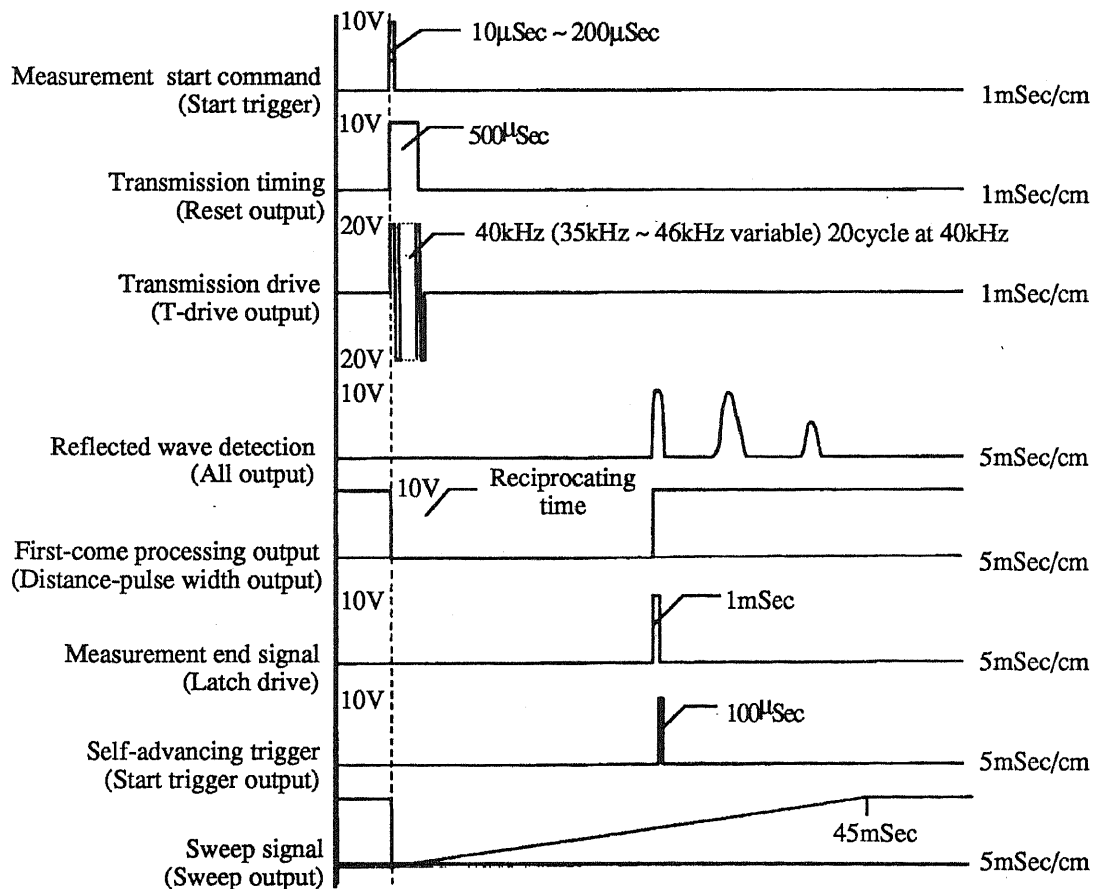


Figure 4.25 Signal data and timing

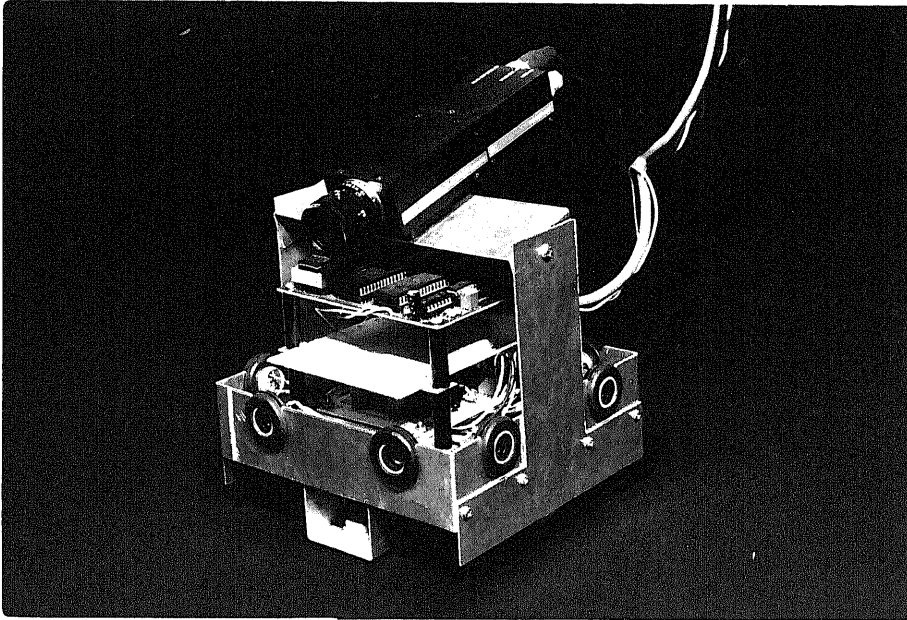


Figure 4.26 Overview of mobile robot

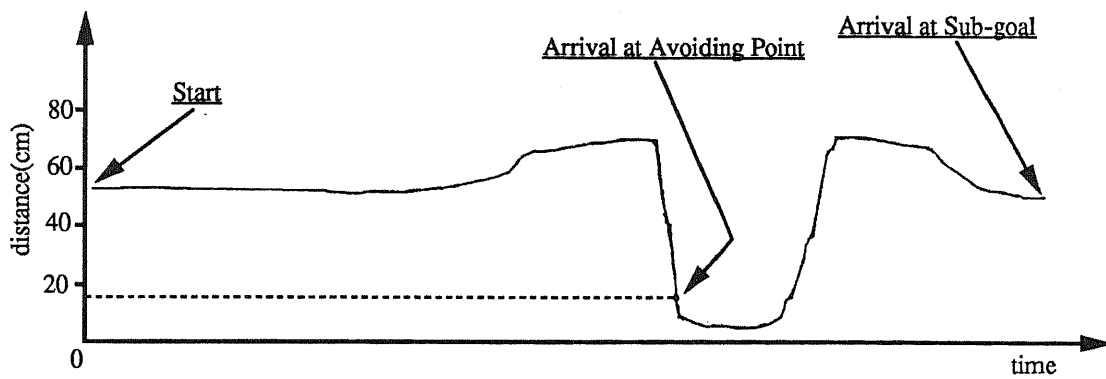


Figure 4.27 Distance data from right-hand ultra-sonic sensor

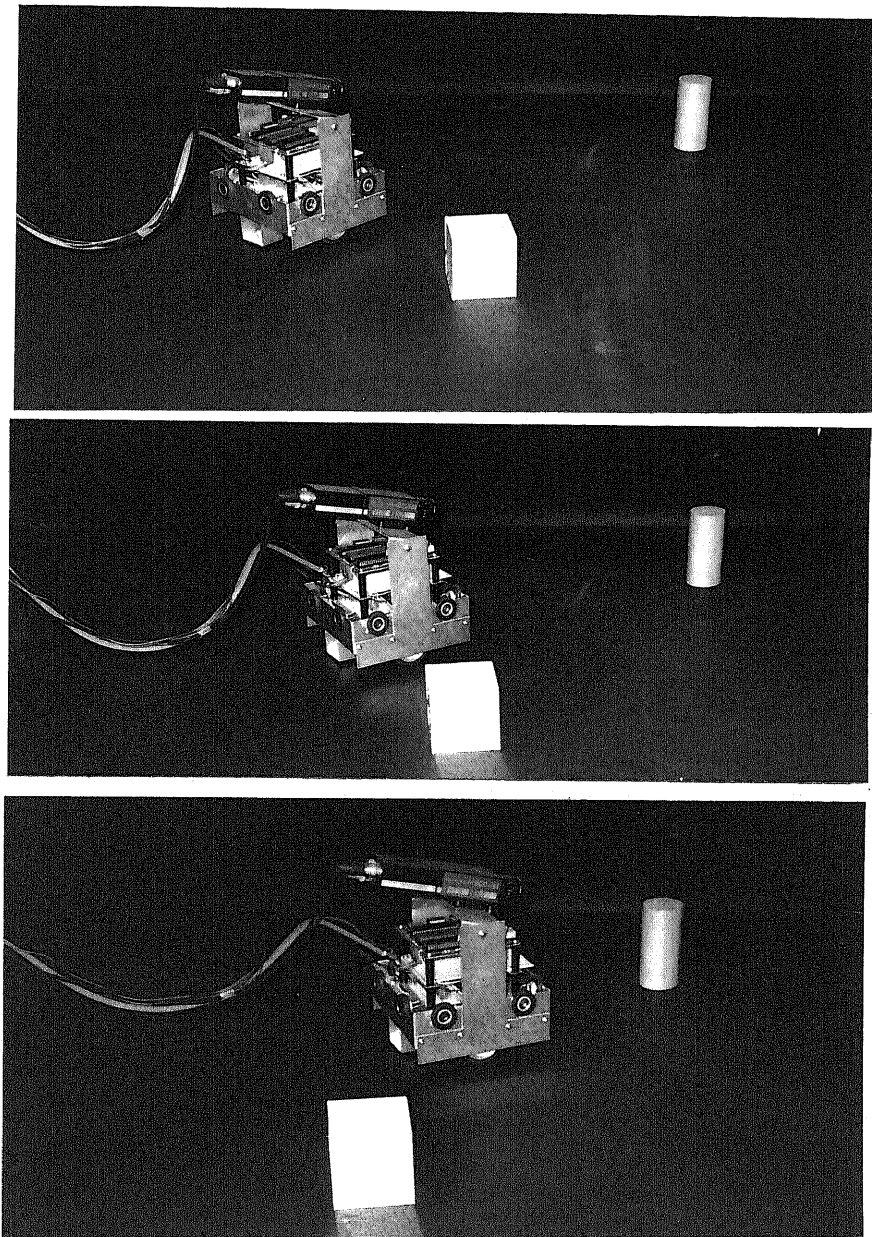


Figure 4.28 Experimental result

## 4.10 まとめ

ロボットが必要な情報を信頼性高く獲得するためにセンサの協調を試みた。まず、視覚情報を用いた移動ロボットのナビゲーション法について述べ、障害物回避のシミュレーションを行った。視覚センサでは、大域的な情報は得やすいが、正確な近接距離情報を得ることは難しい。そこで、超音波センサを用いて近接情報を取り入れる。実際に視覚センサ情報とレンジセンサの協調方法を提案し、シミュレーションによってその有効性を示した。さらに、実験用移動ロボットを用いて、視覚センサと超音波センサの協調による実験を行い、本手法の実験的検証を行った。

提案した融合方法では、視覚情報を距離情報に直して融合しており、大域的な情報である視覚情報を十分に利用しているとはいえない。異種センサ情報の融合に関する新しい手法が望まれる。

## 参考文献

- [有本] 有本, "カルマン・フィルタ," 産業図書.
- [Brady88] J.M.Brady, "Special Issue on Sensor Data Fusion," Int. Journal of Robotics Research, 7-6, pp.2-4 (1988).
- [Chiu86] S.L.Chiu, D.J.Morley, J.F.Martin, "Sensor Data on a Parallel Processor," Proc. of IEEE Int. Conf. on Robotics and Automation, pp.1629-1633 (1986).
- [Harmon86] S.Y.Harmon, G.L.Bianchini, B.E.Pinz, "Sensor Data Fusion through a Distributed Blackboard," Proc. of IEEE Int Conf. on Robotics and Automation, pp.1449-1454 (1986).
- [Henderson85] T.Henderson, "The Specification of Logical Sensors," Proc. of IEEE Int. Workshop on Intelligent Control, pp.102-109 (1985).
- [比留川85] 比留川, 北村, "学習制御系と適応プロダクションシステム," 第27回自動制御連合講演会, pp.59-60 (1985).
- [石川88] 石川, "センサフュージョンシステム (感覚情報の統合メカニズム)," 日本ロボット学会誌, Vol.6, No.3, pp.79-83 (1988).
- [石川90] 石川, "センサフュージョンの現状と課題," 第29回計測自動制御学会学術講演会, pp.1-8 (1990).
- [亀島87] 亀島, 小川, 中野, "再帰型画像処理機構を用いた移動ロボットの視覚誘導," 日本ロボット学会誌5巻5号, pp.11-18 (1987).
- [片山] 片山, "応用カルマンフィルタ," 朝倉書店.
- [木下90] 木下, "センサ情報の統融合化に関する研究の現状," 日本ロボット学会誌8巻6号, pp.88-94 (1990).
- [久保田87] 久保田, 橋本, 原島, "移動ロボットにおけるセンサ情報処理," 第26回SICE学術講演会予稿集, pp.519-520 (1987).
- [Kubota87] T.Kubota, H.Hashimoto, F.Harashima, "Control of Mobile Robot Based on Cooperation of Vision and Range Sensors," Proc. of Int. Workshop on Robotics : Trends, Technology and Applications, pp.197-206 (1987).

- [Kubota88] T.Kubota, H.Hashimoto, F.Harashima, "Path Searching of Mobile Robot Based on Cooperation of Sensors," Proc. of IEEE Int. Workshop on Intelligent Robots and Systems, pp.569-574 (1988).
- [久保田 89] 久保田, 橋本, 原島, "視覚センサとレンジセンサとの協調による移動ロボットのナビゲーション," 日本ロボット学会誌, Vol.7, No.4, pp.275-283 (1989).
- [Luo86] R.C.Luo, M.H.Lin, R.S.Scherp, "Multi-Sensor Based Intelligent Robot System," Proc. of IEEE Int. Conf. on Industrial Electronics, pp.238-243 (1986).
- [Luo87] R.C.Luo, M.H.Lin, R.S.Scherp, "The Issues and Approaches of a Robot Multi-Sensor Integration," Proc. of IEEE Int. Conf. on Robotics and Automation, pp.1941-1946 (1987).
- [Luo88] R.C.Luo, M.H.Lin, R.S.Scherp, "Dynamic Multi-Sensor Data Fusion System for Intelligent Robots," IEEE Journal of Robotics and Automation, Vol.RA-4, No.4, pp. (1988).
- [Luo89] R.C.Luo, M.G.Kay, "Multisensor Integration and Fusion in Intelligent Systems," IEEE Trans. on Systems, Man, and Cybernetics, Vol.19, No.5 (1989).
- [村松 87] 村松, 菅沼, 伊藤, "障害物回避における知識の利用," 第6回知識工学シンポジウム, pp.33-36 (1987).
- [南方 86] 南方, 浅田, 辻, "到達目標の視野指定と視覚フィードバックを用いた移動," 第4回日本ロボット学会学術講演会予稿集, pp.349-350 1986
- [Shafer86] S.A.Shafer, A.Stentz, C.E.Thorpe, "An Architecture for Sensor Fusion in a Mobile robot," Proc. of IEEE Int. Conf. on Robotics and Automation, pp.2002-2011 (1986).
- [鈴木 87] 鈴木, 大西, 有本, "連想データベース自己組織化と自律移動知能ロボットへの応用," 日本ロボット学会誌5巻6号, pp.3-13 (1987).
- [高橋 90] 高橋, 石川, "物理ネットワークによる内部表現を用いたセンサフュージョンシステム," 計測自動制御学会論文集, Vol.26. No.7, pp.803-810 (1990).
- [Wallance85] R.Wallance, A.Stentz, C.E.Thorpe, H.Moravec, W.Whittaker, T.Kanade, "First Results in Robot Road Following," Proc. of Int. Joint Conf. on Artificial Intelligence, pp.1089-1095 (1985).

[山本87] 山本, "移動ロボットのビジュアルコントロールシステムの個性と障害物回避に関する研究," 東京大学大学院修士論文 (1987).

[Zheng89] Y.F.Zheng, "Integration of Multiple Sensors into a Robotic System and its Performance Evaluation," IEEE Trans. on Robotics and Automation, Vol.5, No.5, pp.658-669 (1989).

## Chapter 5

# 動的環境における 移動ロボットの行動制御

本章では、複数の移動障害物が存在する環境（動的環境）において走行する移動ロボットに対し、人間と同じように状況に応じて柔軟な行動決定を行う能力を付与するために、ファジィ推論を用いて衝突危険度を評価し、ルールに基づき移動ロボットの衝突回避行動を形成する手法について述べる。



## 5.1 はじめに

人間はあいまいな環境情報により障害物の動作予測をしながら、柔軟な回避を行っている。このような的確な状況判断に基づく人間の回避アルゴリズムを模擬する試みは、各種移動体の制御において重要な研究分野の一つとなっている。一方、ロボット技術の進歩によって、複数台の移動ロボットが同一環境内で同時に作業するような状況、あるいは人間と共存して移動するような状況は十分に考え得ることである [斉藤 88,90]。このような状況において、固定された障害物のみならず移動する障害物に対して衝突回避を行うことが必要不可欠である。移動ロボットが移動する物体を回避しつつ移動を行うためには、障害物の移動状態を認識し、それに基づいて衝突回避のための移動計画をたてる必要がある。従来より障害物回避に関する研究 [Koch85][久保田 87] はかなりなされているが、主に固定障害物に対するものであった。最近になって、移動障害物を対象とした衝突回避手法 [Kant88][Freund88][Fujimura89][今野 89] が提案されている。

移動ロボットの障害物回避問題においては、従来の代表的なものに、迷路脱出法、侵入禁止領域法、人工ポテンシャル法 [Khatib86]、コンフィギュレーション空間法、ヒューリスティック手法 [Hashimoto88][鈴木 90] などがある。特に、人間の回避特性などを記述した柔軟な知的制御を実現しようとした場合、ヒューリスティック手法が有効であると考えられる。武野ら [武野 86] はヒューリスティック関数を定義し、衝突危険性をチェックしながら将来の回避経路を決定している。また、前田ら [前田 88] によって人間の回避行動を状況判断による危険度認識と回避動作を決定する行動決定の2つのフェーズで構成した手法も提案されている。しかしながら、1つの障害物のみを扱っているものや、障害物の位置及び速度が得られているという条件を設けているものが多い。

本研究では、動的な障害物が存在する環境において走行する移動ロボットに対し、人間と同じようなフレキシブルな状況判断能力をもたせ、知的な回避行動を実現することを目的とする。人間の場合を考えてみると、移動時に物体が近づいたとき、潜在的に危険を察知する能力をもっており、大まかな状況判断を瞬時に行う。このような見地から、直感的な状況判断をファジィ推論により求める。ここでは、複数の移動障害物が存在する環境（動的環境）を対象とし、視覚情報を用いた移動ロボットの障害物回避方法について述べる。まず、画像データを処理することにより、移動物体に対する位置・速度情報を得る。次にこれらの情報を基にファジィ推論を用いて衝突の危険

度を評価する。最後にプロダクションルールにより回避行動が自動生成される。

## 5.2 問題の設定

本研究では、作業空間として屋内を想定し、単眼視による環境観測を行う。一般に、静止した1台のカメラの画像から、奥行き情報は得られないが、「環境内の物体は床面上に立っていてその最下部がみえている」という発見的知識を用いて、物体の位置や大きさを算定する [Kubota88]。

前提条件として、以下のものを設定する。

- 移動ロボットの進むべきゴールの位置はあらかじめ与えられており、視覚によって認識できる。
- 移動ロボットはほぼ水平な床面上を動く。
- 障害物に関しては、まわりの環境となんらかの色、明暗の変化があり（視覚的に検出可能）、また連続的な運動をする。
- 移動ロボットは進行方向に対して真横より前方の範囲に存在する障害物に対してのみ回避行動を行う。

提案する障害物回避方法 [Kubota90] は、Figure 5.1 に示すように4つのステップからなる。第1ステップでは、視覚センサを用いて移動物体の検出を行う。得られた画像データを処理することにより移動障害物の相対位置及び相対速度情報を得る。第2ステップでは、ロボットと障害物の危険度を認識する。第3ステップでは、衝突危険度から回避行動を形成する。最後に第4ステップで、移動ロボットのステアリング及び速度の制御が行われる。

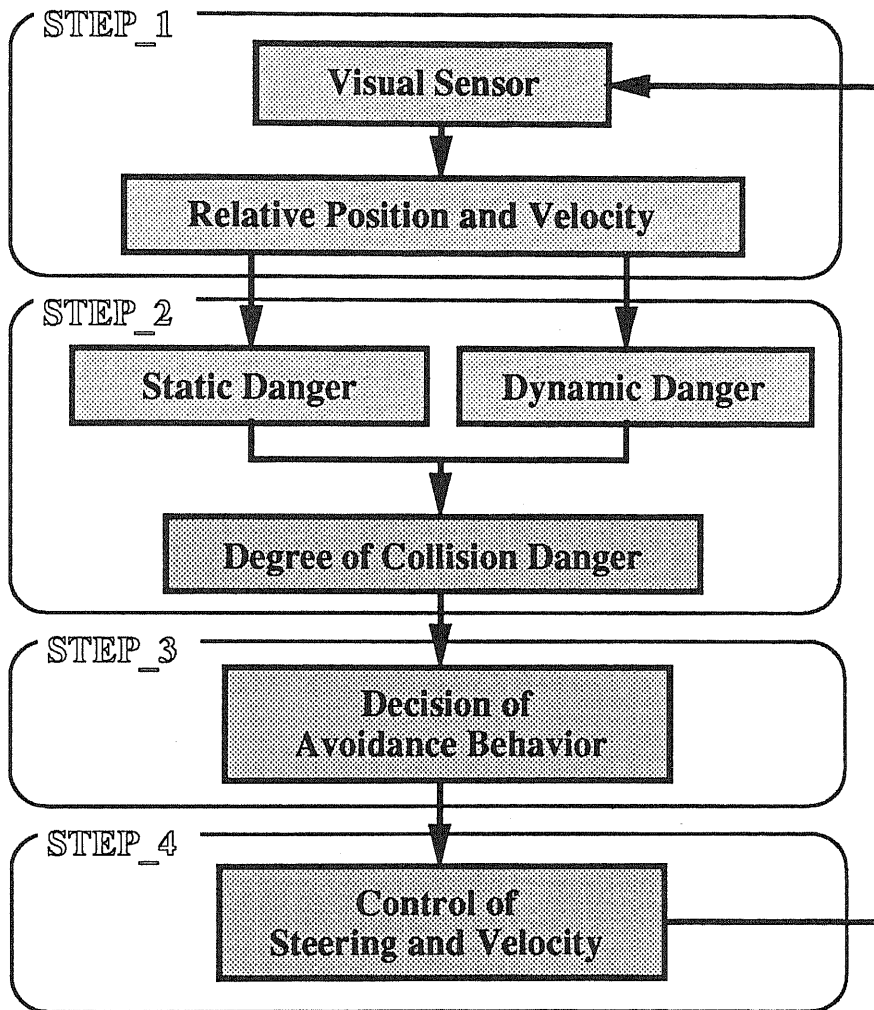


Figure 5.1 Flow chart of obstacle avoidance method

## 5.3 視覚センサによる障害物の検出

### 5.3.1 仮想注目点の設定

物体の3次元運動を認識するために動画像処理がよく用いられており、オプティカルフローなどを用いて動物体の位置・運動の情報を得る手法 [Rives85][Tsuji85][八木86][藤原90] が提案されている。本研究では、画像処理に負担をかけずにリアルタイム処理を行うという観点から、移動物体に対し仮想注目点 [久保田89] を導入することによって、ロボットからみた移動物体のおおまかな運動情報を得る。Figure 5.2 に示すように物体  $O_k$  の特徴点として床面上の3点  $A_k(a_{kx}, a_{ky})$ 、 $B_k(b_{kx}, b_{ky})$ 、 $C_k(c_{kx}, c_{ky})$  を用いる。

まず、ロボットからみた動物体の相対位置情報を得るために画像をサーチして特徴点3点を検出する。この3点から仮想注目点  $P_k$  を次のように設定する。

$\text{IF } a_{kx} \geq m \quad \text{THEN } P_k = (a_{kx}, c_{ky})$ $\text{IF } b_{kx} < m \quad \text{THEN } P_k = (b_{kx}, c_{ky})$ $\text{ELSE } P_k = (\frac{1}{2}(a_{kx} + b_{kx}), c_{ky})$
--

### 5.3.2 障害物の運動情報の推定

ここでは、障害物の運動情報は Figure 5.3 に示すようにロボットの極座標表示で表す。移動障害物（注目点）の相対位置情報は、ロボットの極座標系（角度は進行方向が0、右が正、左が負）からみた障害物の距離  $L$  と方位  $\theta$  で表し、相対速度情報は、障害物の方向を向いたロボット極座標系（角度は障害物の方向が0、右が正、左が負）からみた障害物の速度  $V$  と速度方位  $\phi$  で表す。相対位置  $(L, \theta)$  は画像から検出された座標値  $P_k(px_k, py_k)$  から計算される。また、相対速度  $(V, \phi)$  は、各画像を処理して位置情報を検出し、その差を計算することによって推定する。このようにして

得られた運動情報はかなりのあいまいさを含むが、障害物回避を行うのには十分と考えられる。あいまいさに関しては、第(5.7)節で考察する。

### A. 相対位置情報

Figure 5.4(b) は、移動ロボットと注目点  $P$  を上からみた図を示している。ロボット座標系は、移動ロボットの位置を原点とし、移動ロボットの進行方向が  $O_y$  軸となるように設定している。この注目点の相対位置  $(L, \theta)$  は、画像から検出された座標値  $P_k(p_{kx}, p_{ky})$  から次のようにして計算される。

いま、Figure 5.4(c) における  $\beta$  は次式で表される。

$$\beta = \beta_0 + \gamma \quad (5.1)$$

$$\gamma = \tan^{-1} \frac{2m - p_{ky}}{F} \quad (5.2)$$

また、画像上で検出される点  $P$  の  $x$  座標値  $p_{kx}$  は次式で表される。

$$p_{kx} = \frac{F \sin \beta}{H \cos \gamma} RX + m \quad (5.3)$$

ただし、

- $\beta_0$  : カメラの傾斜角
- $F$  : 焦点距離
- $H$  : カメラの床面からの高さ

これらの式を用いると

$$RX = \frac{(p_{kx} - m)H \cos \gamma}{F \sin \beta} \quad (5.4)$$

$$RY = H \cot \beta \quad (5.5)$$

などの関係式が得られる。

ところで、 $(L, \theta)$  は次の式で表される。

$$L = \sqrt{RX^2 + RY^2} \quad (5.6)$$

$$\theta = \tan^{-1} \frac{RX}{RY} \quad (5.7)$$

式(5.4)、式(5.5)を式(5.6)、式(5.7)に代入して

$$L = \sqrt{H^2 \cos^2 \beta + \frac{(p_{kx} - m)^2 H^2 \cos^2 \gamma}{F^2 \sin^2 \beta}} \quad (5.8)$$

$$\theta = \tan^{-1} \frac{(p_{kx} - m) \cos \gamma}{F \cos \beta \sin \beta} \quad (5.9)$$

を得る。

## B. 相対速度情報

移動物体との相対速度情報を得るために、ここでは3枚の連続画像を用いる。各画像を処理して位置情報を検出し、その差を計算することによって相対速度を推定する。相対速度としては Figure 5.3 に示すような極座標表示  $(V, \phi)$  を用いる。

いま、相対速度を  $U_0 = (U_{0x}, U_{0y})$  とする。

$$U = R_\theta \cdot U_0 = [U_x, U_y]^T \quad (5.10)$$

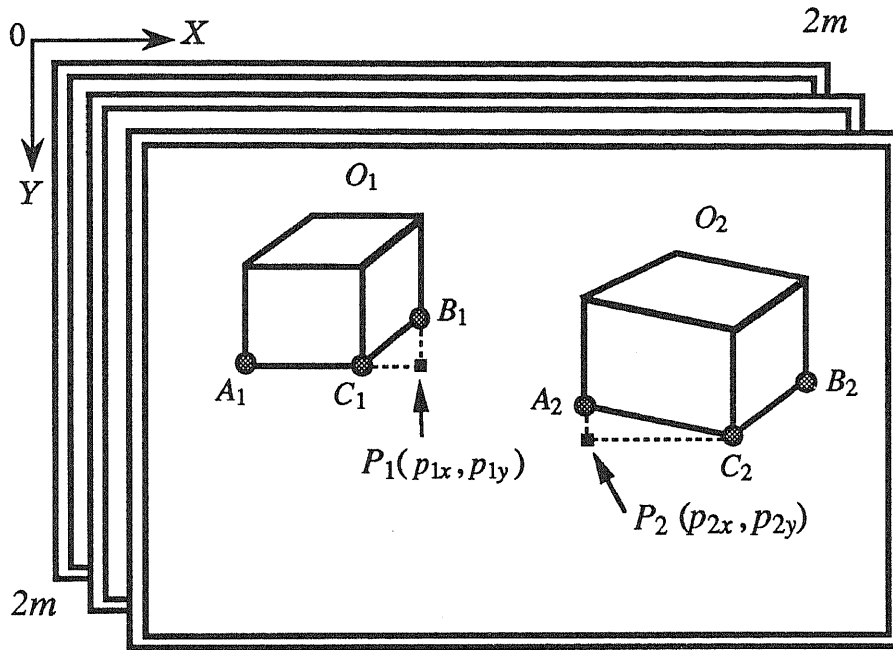
$$R = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \quad (5.11)$$

ここで、 $\theta$  は障害物の方向を表している。

障害物の相対速度  $(V, \phi)$  は次の式を用いて計算される。

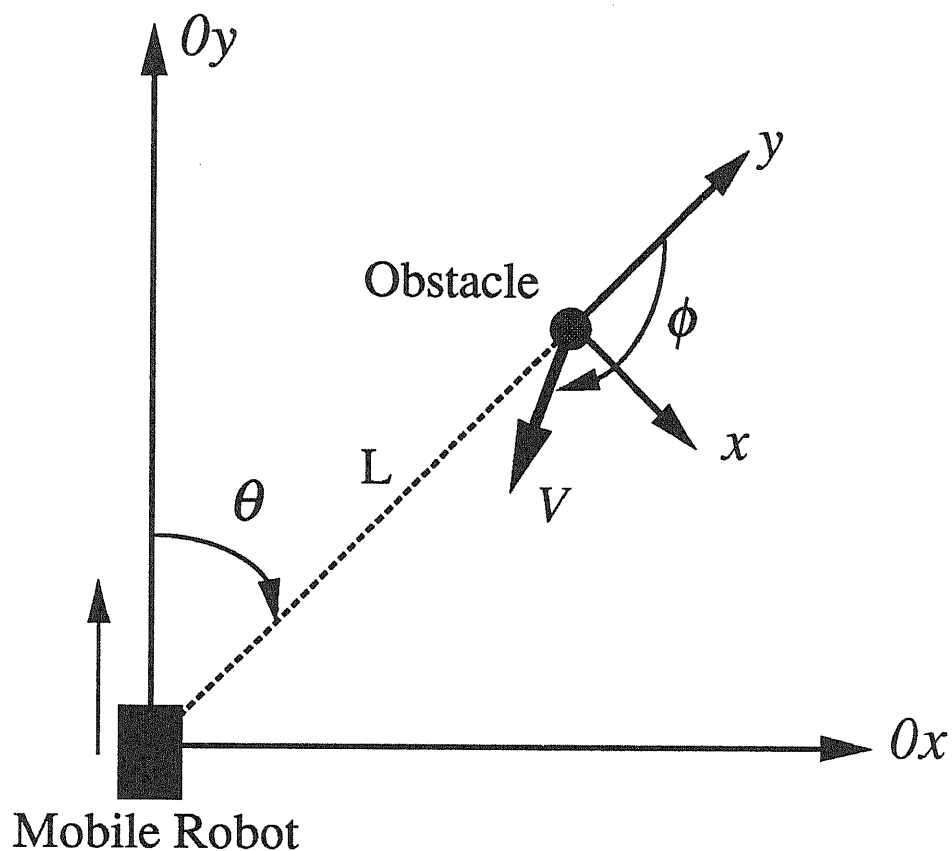
$$V = \sqrt{U_x^2 + U_y^2} \quad (5.12)$$

$$\phi = \frac{\pi}{2} \cdot \tan^{-1} \frac{U_y}{U_x} \quad (5.13)$$



IF  $a_{kx} \geq m$  THEN  $P_k = (a_{kx}, c_{ky})$   
 IF  $b_{kx} < m$  THEN  $P_k = (b_{kx}, c_{ky})$   
 ELSE  $P_k = (\frac{1}{2}(a_{kx} + b_{kx}), c_{ky})$ .

Figure 5.2 Feature points of obstacles



Relative position  
( $L, \theta$ )  
Relative velocity  
( $V, \phi$ )

Figure 5.3 Robot coordinate system



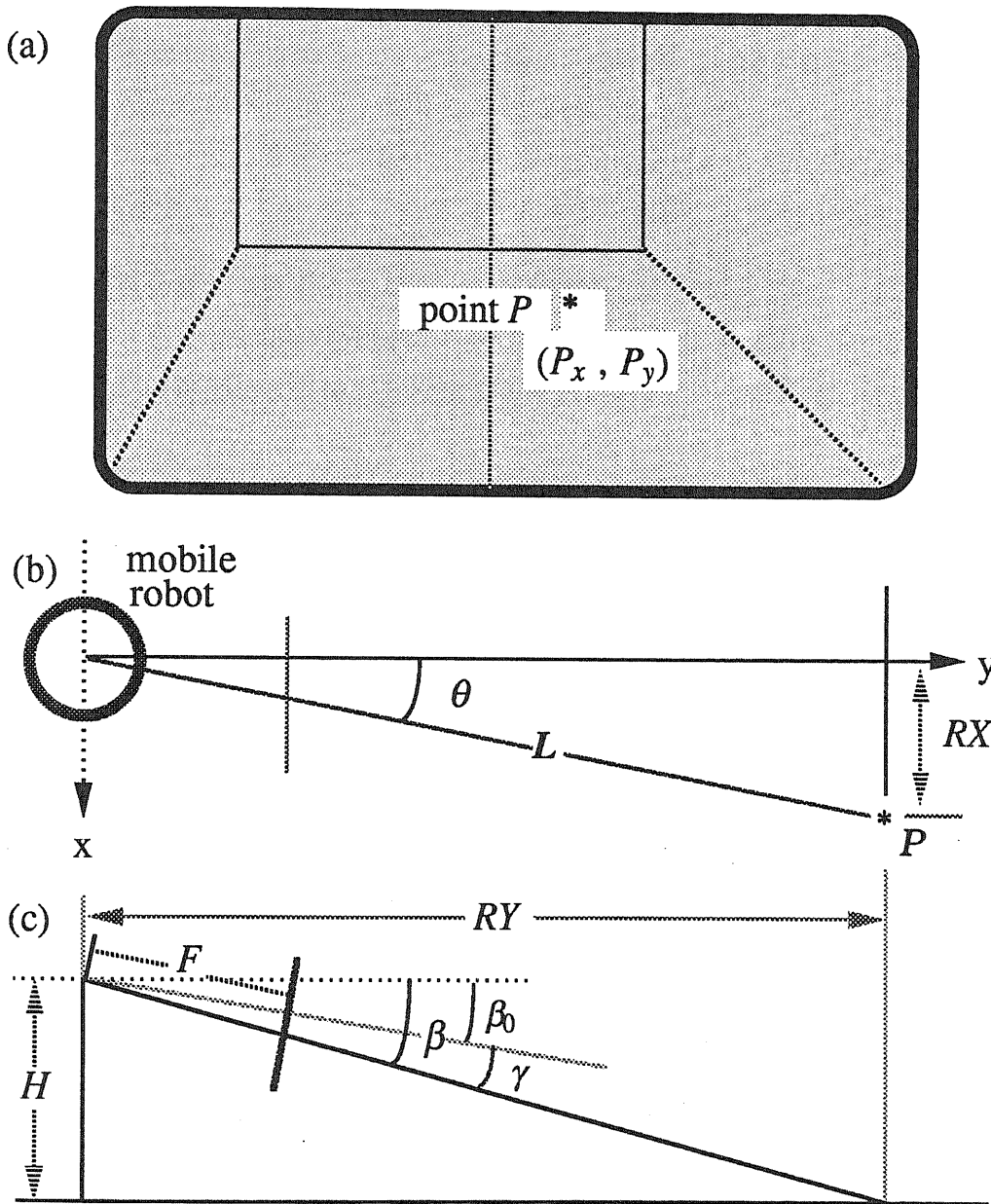


Figure 5.4 Camera system and image

## 5.4 衝突危険度による状況判断

人間が障害物を回避することを考えた場合、各場面で正確な計測をしているとは考えられず、周囲の状況の捉え方はかなり感覚的である。そこで、直感的な状況把握を試みる。ファジィ推論を用いて静的及び動的危険度を求める方法が前田・竹垣ら [前田 88] によって提案されている。ここではその手法を拡張して衝突危険度を評価する。

まず、視覚情報から得られたロボットと障害物の相対位置情報  $(L, \theta)$  を用いて静的危険度  $\varepsilon$  を、また相対速度情報  $(V, \phi)$  から動的危険度  $\delta$  を求める。これら2つの危険度からファジィ推論により衝突危険度を  $\sigma$  を求める。

### 5.4.1 静的危険度

ファジィを用いてロボットと障害物の相対的な位置関係から生じる危険性を表現する。障害物までの距離  $L$  は、0 から  $L_0$  までに存在するときを考え、Figure 5.5(a) に示すような3段階のメンバシップ関数を設定する。ただし、 $L_0$  はロボットの動く環境に応じて決定される。 $N$  は近いとき、 $M$  は中間域、 $F$  は遠いときを示している。一方、障害物の方向  $\theta$  は移動ロボットの進行方向に対して真横より前方の範囲 ( $-\frac{1}{2}\pi$  から  $\frac{1}{2}\pi$  まで) に物体が存在するときを考え、Figure 5.5(b) に示すような4段階のファジィ表現を設定する。ロボットに対して、 $LB$  は左側に大、 $LS$  は左側に小、 $RS$  は右側に小、 $RB$  は右側に大の角度にそれぞれ物体が存在するときを表している。

次に、静的危険度  $\varepsilon$  を Figure 5.6 に示すメンバシップ関数でファジィ表現を行う。ここで、左が大変危険 ( $LDB$ )、やや危険 ( $LDS$ )、やや安全 ( $LSS$ )、安全 ( $LSB$ )、右が大変危険 ( $RDB$ )、やや危険 ( $RDS$ )、やや安全 ( $RSS$ )、安全 ( $RSB$ ) の8つのファジィラベルで表す。メンバシップ関数は横軸の負側がロボット左側の危険度、正側がロボット右側の危険度を示し、0 付近が安全、 $-1$  と  $+1$  付近が危険であることを表している。以上のメンバシップ関数を用いて Figure 5.7 に示すルールに従って、相対位置情報のファジィ状態に対して静的危険度のファジィ認識を行う。前件部では Figure 5.5(a),(b) より距離、方向のファジィ表現をおこない、これに対し後件部では、Figure 5.6 より静的危険度のファジィ表現の1つを対応させる。たとえば、

IF 「距離が近く ( $N$ ) て方向がやや左 ( $LS$ ) に障害物が存在する」  
 THEN 「左が大変危険 ( $LDB$ )」

のように判断する。同様にして計 12 の状態をファジイルールで表現できる。以上のルールを用いてファジィ推論を行うと距離と方向による適合度で重みづけされた静的危険度のメンバシップ関数が得られ、これらの min-max 合成後、CG法により重心位置の危険度がそのときの静的危険度  $\epsilon$  として決定される。

### 5.4.2 動的危険度

ここでは、移動ロボットと障害物との相対速度から生じる危険性をファジィで表現する。極座標速度  $V$  のメンバシップ関数を Figure 5.8(a) に示す。速度は、0 から  $V_0$  までの大きさを考え、2 段階のメンバシップ関数を設定する。ただし、 $V_0$  はロボットの動く環境に応じて決定される。 $VS$  は遅いとき、 $VF$  は速いときを示している。Figure 5.8(b) は、障害物の速度方向のメンバシップ関数を示す。速度方向  $\phi$  は  $-\pi$  から  $\pi$  までを考え、6 段階のファジィ表現を設定する。ロボットに対して、 $VLB$ 、 $VLM$ 、 $VLS$  は左側、 $VRS$ 、 $VRM$ 、 $VRB$  は右側の速度方向を示す。

次に、動的危険度  $\delta$  のメンバシップ関数を Figure 5.9 に示す。静的危険度と同じように、左が大変危険 ( $LDB$ )、やや危険 ( $LDS$ )、やや安全 ( $LSS$ )、安全 ( $LSB$ )、右が大変危険 ( $RDB$ )、やや危険 ( $RDS$ )、やや安全 ( $RSS$ )、安全 ( $RSB$ ) の 8 つのファジィラベルで表す。同様に、以上のメンバシップ関数を用いて Figure 5.10 に示すルールに従って、相対速度情報のファジィ状態に対して動的危険度のファジィ推論を行う。

### 5.4.3 衝突危険度の評価

ここでは、静的危険度と動的危険度の 2 つの危険度から、さらにファジィ推論により衝突危険度  $\sigma$  を求める。 $\sigma$  は、Figure 5.11 に示すように左が大変危険 ( $LDB$ )、やや危険 ( $LDS$ )、やや安全 ( $LSS$ )、安全 ( $LSB$ )、右が大変危険 ( $RDB$ )、やや危険 ( $RDS$ )、やや安全 ( $RSS$ )、安全 ( $RSB$ ) の 8 つのファジィラベルで表す。ファジィ制御ルールを Table 5.1 に示す。たとえば、

IF  $\varepsilon$  is LDB and  $\delta$  is RDB  
THEN  $\sigma$  is LDB

となる。これらのルールを用いて min-max 合成後、CG法により重心位置の危険度がそのときの衝突危険度として決定される。

#### 5.4.4 ファジィ推論

本手法では、マムダニのファジィ推論を用いており推論方法 [菅野] を簡単に説明する。Figure 5.12 にルールが2つの場合の推論過程を示している。推論手順は以下のようになる。

(1) 入力  $L^*, \theta^*$  に対して、各ルールの前件部の適合度を求める。

ルール1の適合度  $w_1 = 0.45 (= 0.75 \wedge 0.45)$

ルール2の適合度  $w_2 = 0.20 (= 0.20 \wedge 0.60)$

(2) 各ルールの推論結果を求める。

ルール1の推論結果は、ルールの後件部のファジィ集合  $RDB$  を  $w_1 = 0.45$  で切った下の部分（斜線部分）となる。

ルール2の推論結果は、ルールの後件部のファジィ集合  $RDS$  を  $w_2 = 0.20$  で切った下の部分（斜線部分）となる。

(3) 各ルールの推論結果を合成し、出力のファジィ集合を求める。

ルール1、ルール2の推論結果を重ね合わせ、メンバシップ値の大きい方をとる。

(4) 最後に出力のファジィ集合から重心を求めることによって出力  $\varepsilon^*$  を得る。

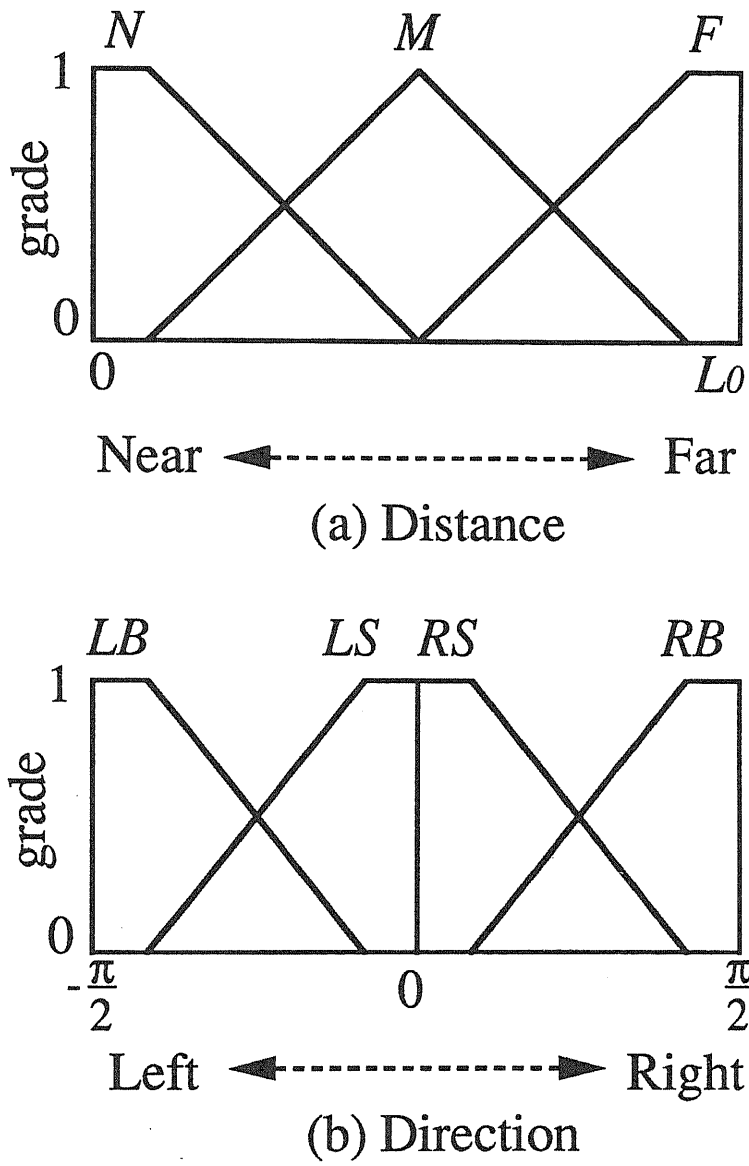


Figure 5.5 Membership function of relative position

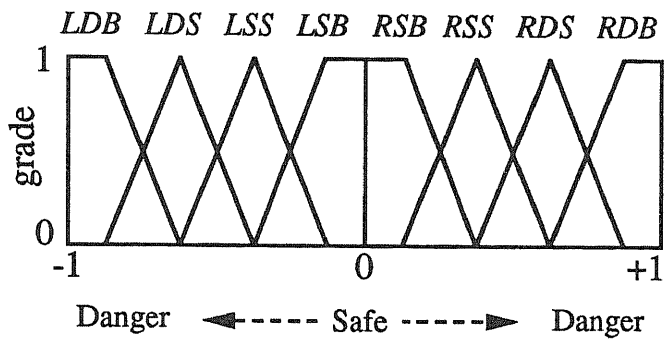


Figure 5.6 Membership function of static danger

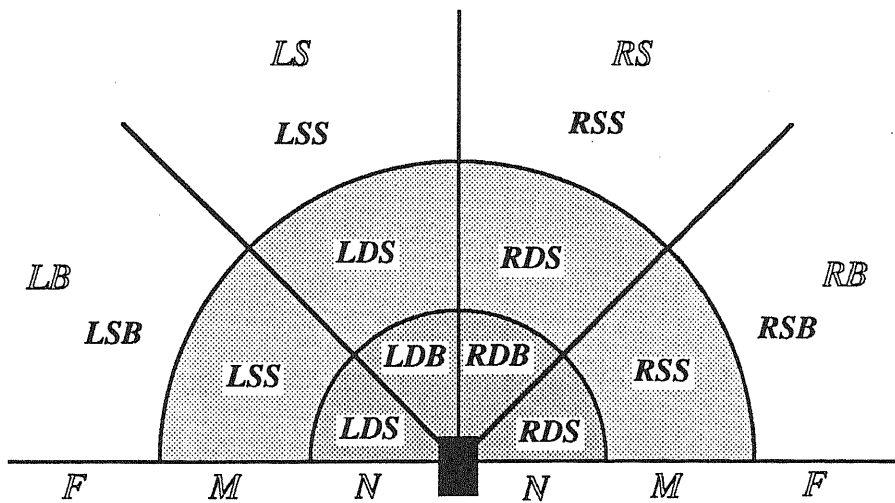


Figure 5.7 Decision of static danger

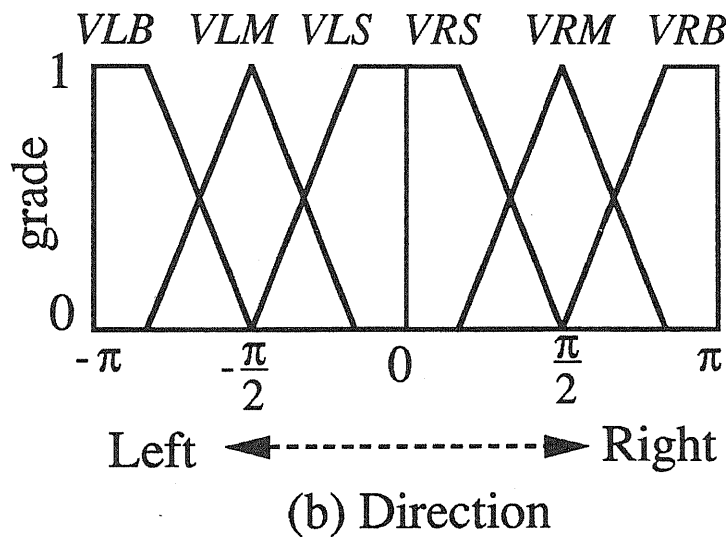
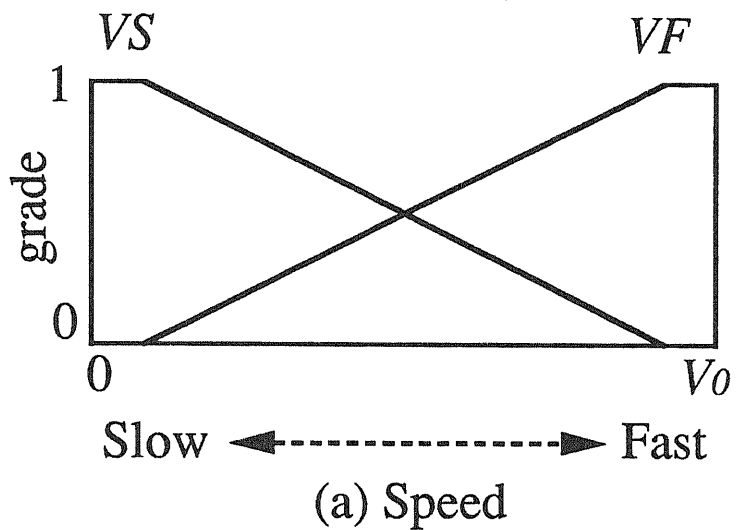


Figure 5.8 Membership function relative velocity

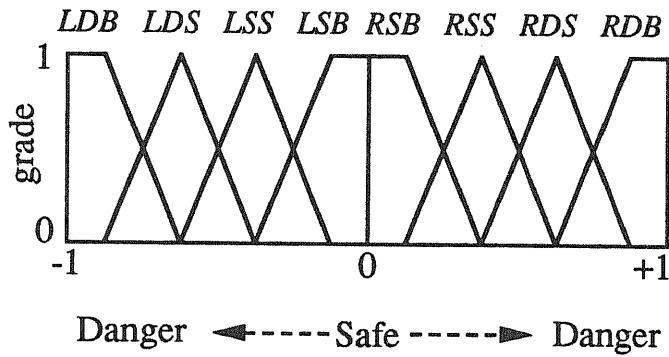


Figure 5.9 Membership function of dynamic danger

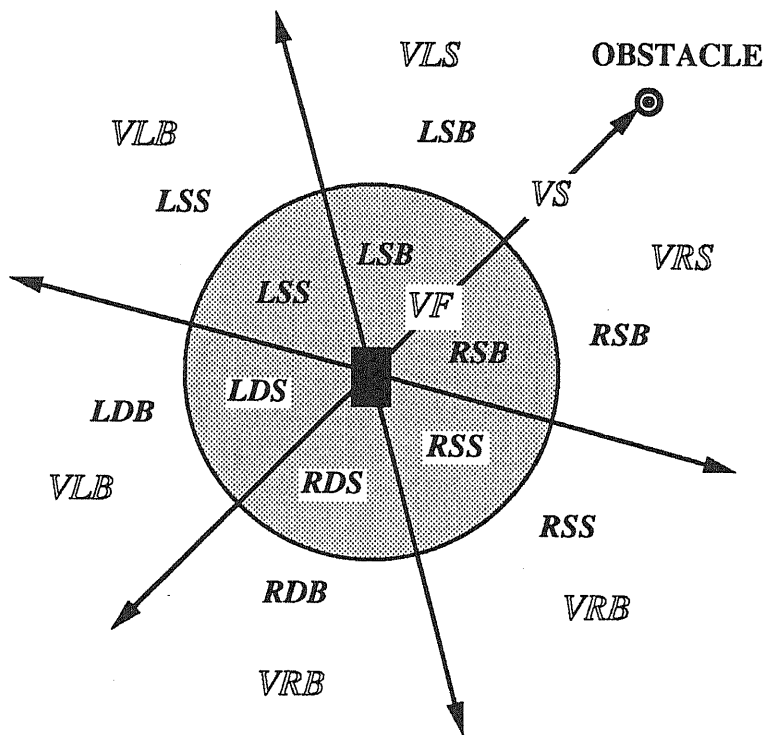


Figure 5.10 Decision of dynamic danger



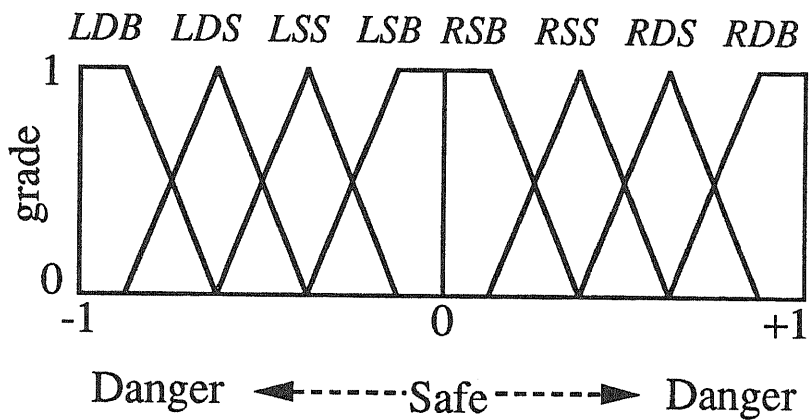


Figure 5.11 Membership function of collision danger

Table 5.1 Decision of collision danger

$\delta \backslash \epsilon$	LDB	LDS	LSS	LSB	RSB	RSS	RDS	RDB
LDB	LDB	LDS	LSS	LSB	RSS	RDS	RDB	RDB
LDS	LDS	LSS	LSB	LSB	RSB	RSS	RDS	RDB
LSS	LSS	LSB	LSB	LSB	RSB	RSB	RSS	RDS
LSB	LSB	LSB	LSB	LSB	RSB	RSB	RSB	RSS
RSB	LSS	LSB	LSB	LSB	RSB	RSB	RSB	RSB
RSS	LDS	LSS	LSB	LSB	RSB	RSB	RSB	RSS
RDS	LDB	LDS	LSS	LSB	RSB	RSB	RSS	RDS
RDB	LDB	LDB	LDS	LSS	RSB	RSS	RDS	RDB

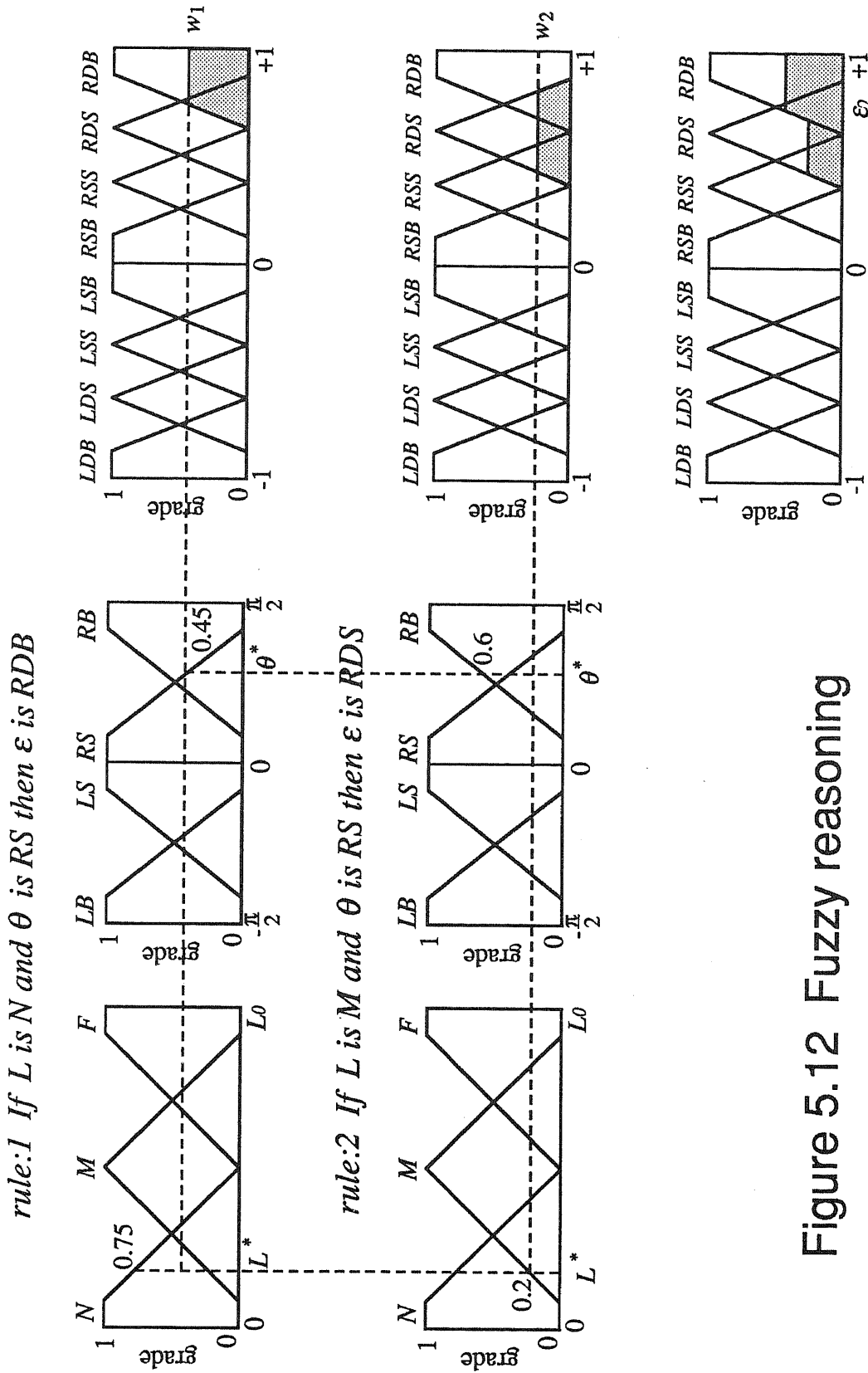


Figure 5.12 Fuzzy reasoning

## 5.5 移動経路の生成

### 5.5.1 回避行動の決定

移動ロボットの障害物回避行動は、プロダクションルールによって決定される。ルールの condition 部分は移動ロボットの周囲の状況を、すなわち衝突危険度及び静的・動的危険度を与え、action 部分は回避方向  $r$  と修正移動速さ  $s$  からなる回避行動 ACT ( $r, s$ ) を与える。ここでは、 $r$  は進行方向に対して右にそれぞれ  $0, \frac{1}{18}\pi, \frac{1}{9}\pi, \frac{1}{6}\pi, \frac{1}{4}\pi, \frac{1}{3}\pi$  の方向の時に  $0, +1, +2, +3, +4, +5$  の値を、同様に左方向の時には  $-1, -2, -3, -4, -5$  の値をとる。 $s$  は速さを表しており、速く (+2)、やや速く (+1)、普通 (0)、やや遅く (-1)、遅く (-2) の5段階とする。また、割り込み行動として停止行動 (00) も与える。( Figure 5.13 参照)

### 回避行動の形成アルゴリズム

1. 障害物  $O_k$  の衝突危険度  $\sigma_k$  及び、 $\epsilon_k$  と  $\delta_k$  をファジィ推論によって求める。
2.  $N$  個の移動物体に対して、 $\{\sigma_1, \sigma_2, \dots, \sigma_N\}$  を評価する。
3.  $|\sigma_i|$  の大きさによって、回避優先順位を設定する。
4. 回避ルール選択テーブルからルール  $AR_j$  を抽出する。
5. ルールの中から回避行動の候補 ACT を取り出す。
6. 危険度状態  $X_{it}(\epsilon_{it}, \delta_{it}, \sigma_{it})$  から  $X_{it+1}(\epsilon_{it+1}, \delta_{it+1}, \sigma_{it+1})$  において、 $|\sigma_{it+1}| \leq |\sigma_{it}|$  となる回避行動を候補の中から決定する。

例えば、移動ロボットは障害物の中で衝突危険度の最も大きい2つの物体のみに注目する。すなわち、衝突危険度  $\sigma_k$  の絶対値の最も大きい物体 ( $k=i$ ) と2番目に大きい物体 ( $k=j$ ) に対して回避行動を生成する。次に、その衝突危険度 ( $\sigma_i, \sigma_j$ ) から Table 5.2 に示されたデシジョンテーブルを用いて回避ルール  $AR$  が抽出される。回避ルール  $AR$  は Table 5.3 に示すようにプロダクションルールで記述されている。このルールには人間の行動パターンを模擬するうえで非常に特徴的な回避戦略を盛り込んでいる。その一例を Table 5.4 に示す。このプロダクションルールを適用することにより回避行動が決定される。

また、回避ルールには、通常 Figure 5.5 に示すように複数の回避行動パターンが選択すべき候補として与えられており、順番によって優先度が割り当てられている。障害物が2台の場合には先頭の回避行動パターンが選ばれるが、3台以上の障害物が注目されたならば、候補のパターンのなかで、そのパターンを選んだとき衝突危険度が最も小さくなると予測されるパターンを選択する。またそのパターンの優先度を更新する。これは、注目する2台以外の障害物が接近したときにローカルな意味での最適な行動を選ぶための戦略であり、また選択された行動の優先度を高めることによって、ある種の学習効果を付加することがねらいとなっている。さらに、人間によって決められたこの回避行動パターンの候補をある評価のもとに変更・追加・削除するなどの機能を実現することも必要と思われる。

### 5.5.2 ナビゲーション

移動ロボットの役割は、スタート地点から出発して障害物に衝突することなく、目標地点に到達することである。ここでは、障害物回避のための回避ベクトルと目標方向を表す目標ベクトルを合成することによって移動ロボットのステアリングを決定する。

従って、移動ロボットの行動戦略は目標到達行動と障害物回避行動とからなる。ここでは、衝突危険度  $\sigma$  を用いて、Figure 5.14 に示すように次の3つのナビゲーション戦略を考える。

(1)  $|\forall\sigma| \leq \sigma_{th1}$  のとき

目標到達行動を優先する。

(2)  $\sigma_{th1} \leq |\exists\sigma| \leq \sigma_{th2}$  のとき

目標到達行動を表す目標到達ベクトルと障害物回避行動を表す回避ベクトルを衝突危険度に基づく重みによって合成することにより、移動ロボットのステアリング方向を決定する。

(3)  $|\exists\sigma| \geq \sigma_{th2}$  のとき

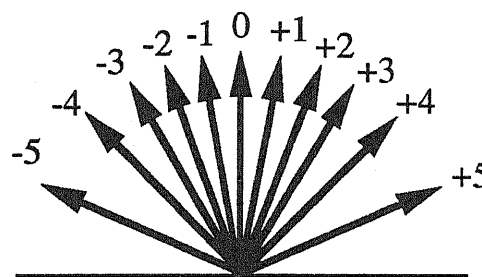
障害物回避行動を優先する。

ただし、 $\sigma_{th1}$  は安全しきい値、 $\sigma_{th2}$  は危険しきい値を表し、その値は作業環境によって決められる。

**Action [ steering, speed ]**

steering command

11 directions



speed command

5 levels

slow	-2
a little slow	-1
medium	0
a little fast	+1
fast	+2

interrupt command

stop      00

Figure 5.13 Action pattern for obstacle avoidance



Table 5.3 Avoidance action patterns

<b>AR0</b> (0,0)	<b>AR2</b> (-1,0)	<b>AR5</b> (+5,0)	<b>AR6</b> (+4,0)	<b>AR8</b> (+5,+1)
<b>AR9</b> (-5,0)	<b>AR10</b> (+1,0)	<b>AR15</b> (-4,0)	<b>AR16</b> (-5,0)	
<b>AR7</b> 0.6 ≤ $\epsilon_{sy}$ < 0.8, -0.8 ≤ $\epsilon_{dt}$ < 0.6 ELSE	(+3,-1) (-4,+1)	<b>AR14</b> 0.6 ≤ $\epsilon_{sy}$ < 0.8, -0.8 ≤ $\epsilon_{dt}$ < 0.6 ELSE	(-3,-1) (+4,+1)	
<b>AR1</b> -1.0 ≤ $\epsilon_{sy}$ < 0.6, $\epsilon_{dj}$ < 0.0 -1.0 ≤ $\epsilon_{sy}$ < 0.6, $\epsilon_{dj}$ ≥ 0.0 -0.6 ≤ $\epsilon_{sy}$ < 0.0	(-1,-1) (-3,0) (-1,0)	<b>AR11</b> 0.6 ≤ $\epsilon_{sy}$ < 1.0, $\epsilon_{dj}$ ≥ 0.0 0.6 ≤ $\epsilon_{sy}$ < 1.0, $\epsilon_{dj}$ < 0.0 0.0 ≤ $\epsilon_{sy}$ < 0.6	(+1,-1) (+3,0) (+1,0)	
<b>AR4</b> 0.2 ≤ $\epsilon_{sx}$ < 0.4, -1.0 ≤ $\epsilon_{dt}$ < -0.8 0.4 ≤ $\epsilon_{sx}$ < 0.6, -1.0 ≤ $\epsilon_{dt}$ < -0.8 0.2 ≤ $\epsilon_{sx}$ < 0.6, -0.8 ≤ $\epsilon_{dt}$ < -0.6 0.6 ≤ $\epsilon_{sx}$ < 0.8, -0.8 ≤ $\epsilon_{dt}$ < -0.6 0.4 ≤ $\epsilon_{sx}$ < 0.8, -0.6 ≤ $\epsilon_{dt}$ < -0.4 0.8 ≤ $\epsilon_{sx}$ < 1.0, -0.6 ≤ $\epsilon_{dt}$ < -0.4 0.6 ≤ $\epsilon_{sx}$ < 1.0, -0.4 ≤ $\epsilon_{dt}$ < -0.2	(-2,0) (-3,0) (-1,0) (-2,0) (-1,0) (-2,0) (-1,0)	<b>AR12</b> -1.0 ≤ $\epsilon_{sx}$ < -0.6, 0.2 ≤ $\epsilon_{dt}$ < 0.4 -1.0 ≤ $\epsilon_{sx}$ < -0.8, 0.4 ≤ $\epsilon_{dt}$ < 0.6 -0.8 ≤ $\epsilon_{sx}$ < -0.4, 0.4 ≤ $\epsilon_{dt}$ < 0.6 -0.8 ≤ $\epsilon_{sx}$ < -0.6, 0.6 ≤ $\epsilon_{dt}$ < 0.8 -0.6 ≤ $\epsilon_{sx}$ < -0.2, 0.6 ≤ $\epsilon_{dt}$ < 0.8 -0.6 ≤ $\epsilon_{sx}$ < -0.4, 0.8 ≤ $\epsilon_{dt}$ < 1.0 -0.4 ≤ $\epsilon_{sx}$ < -0.2, 0.8 ≤ $\epsilon_{dt}$ < 1.0	(1,0) (2,0) (1,0) (2,0) (1,0) (3,0) (2,0)	
<b>AR3</b> a. -1.0 ≤ $\epsilon_{sy}$ < 0.6, $\epsilon_{dj}$ < 0.0 b. -1.0 ≤ $\epsilon_{sy}$ < 0.6, $\epsilon_{dj}$ ≥ 0.0 c. -0.6 ≤ $\epsilon_{sy}$ < 0.0	$\epsilon_{dt}$ < 0.8 $\epsilon_{dt}$ < 0.8 $\epsilon_{dt}$ < 0.6	a. b. c.	(-2,-1) (-3,0) (-2,1)	(-2,0) (-2,1) (-1,0)
	0.6 ≤ $\epsilon_{sx}$ < 0.8, -0.8 ≤ $\epsilon_{dt}$ < -0.6 0.4 ≤ $\epsilon_{sx}$ < 0.8, -0.6 ≤ $\epsilon_{dt}$ < -0.4 0.8 ≤ $\epsilon_{sx}$ < 1.0, -0.6 ≤ $\epsilon_{dt}$ < -0.4 0.6 ≤ $\epsilon_{sx}$ < 1.0, -0.4 ≤ $\epsilon_{dt}$ < -0.2		(-2,-1) (-3,0) (-2,-1) (-1,+1)	(-2,0) (-1,0) (-2,0) (-1,0)
<b>AR13</b> a. 0.6 ≤ $\epsilon_{sy}$ < 1.0, $\epsilon_{dj}$ ≥ 0.0 b. 0.6 ≤ $\epsilon_{sy}$ < 1.0, $\epsilon_{dj}$ < 0.0 c. 0.0 ≤ $\epsilon_{sy}$ < 0.6	$\epsilon_{dt}$ < 0.4 $\epsilon_{dt}$ < 0.6 $\epsilon_{dt}$ < 0.6	a. b. c.	(+1,-1) (+2,-1) (+1,-1)	(+3,0) (+3,0) (+3,0)
	-0.8 ≤ $\epsilon_{sx}$ < -0.6, 0.6 ≤ $\epsilon_{dt}$ < 0.8 -0.6 ≤ $\epsilon_{sx}$ < -0.2, 0.6 ≤ $\epsilon_{dt}$ < 0.8 -0.6 ≤ $\epsilon_{sx}$ < -0.4, 0.8 ≤ $\epsilon_{dt}$ < 1.0 -0.4 ≤ $\epsilon_{sx}$ < -0.2, 0.8 ≤ $\epsilon_{dt}$ < 1.0		(+2,-1) (+1,-1) (+2,-2) (+2,-1)	(+3,0) (+3,0) (+3,0) (+3,0)

Table 5.4 Avoidance action

<i>AR3</i>		a.	b.	c.
a. $-1.0 \leq \epsilon_{sj} < -0.6, \epsilon_{dj} < 0.0$	$\epsilon_{di} < -0.8$	(-2,-1)	(-3,0)	(-2,0)
b. $-1.0 \leq \epsilon_{sj} < -0.6, \epsilon_{dj} \geq 0.0$	$\epsilon_{di} < -0.8$	(-2,-2)	(-3,0)	(-2,1)
c. $-0.6 \leq \epsilon_{sj} < 0.0$	$\epsilon_{di} < -0.6$	(-1,-1)	(-3,0)	(-1,0)
	$0.6 \leq \epsilon_{si} < 0.8, -0.8 \leq \epsilon_{di} < -0.6$	(-2,-1)	(-3,0)	(-2,0)
	$0.4 \leq \epsilon_{si} < 0.8, -0.6 \leq \epsilon_{di} < -0.4$	(-1,-1)	(-3,0)	(-1,0)
	$0.8 \leq \epsilon_{si} < 1.0, -0.6 \leq \epsilon_{di} < -0.4$	(-2,-1)	(-3,0)	(-2,0)
	$0.6 \leq \epsilon_{si} < 1.0, -0.4 \leq \epsilon_{di} < -0.2$	(-1,+1)	(-3,0)	(-1,0)

Table 5.5 An example of action rule

<i>ARI</i>	
$-1.0 \leq \epsilon_1 < -0.6, \delta > 0.0$	(-1,-1), (-2,0), (-2,-1)
$-1.0 \leq \epsilon_1 < -0.6, \delta \geq 0.0$	(-3,0), (-4,0)
$-0.6 \leq \epsilon_1 < 0.0$	(-1,0), (-2,0), (-2,-1)



$$(1) \quad \forall |\sigma| \leq \sigma_{th1} \quad \begin{matrix} \cdot \text{ safety} \\ \cdot \text{ threshold} \end{matrix}$$

*goal-following* **Behavior**



$$(2) \quad \exists |\sigma| \geq \sigma_{th2} \quad \begin{matrix} \cdot \text{ danger} \\ \cdot \text{ threshold} \end{matrix}$$

*avoidance* **Behavior**



$$(3) \quad \sigma_{th1} \leq \exists |\sigma| \leq \sigma_{th2}$$

*goal following + avoidance* **Behavior**

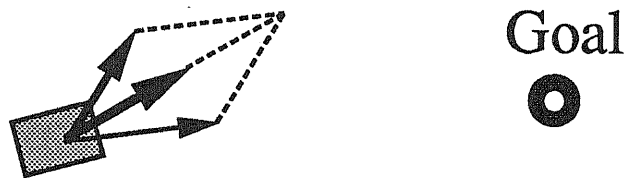


Figure 5.14 Navigation strategy

## 5.6 シミュレーション

3次元グラフィックスを用いて、移動ロボットが視覚センサによって取り込む画像をシミュレートした。一般に、画像は投影変換によって表されるので、対象をフレームモデルによって表す。ここでは、ゴールの位置情報を既知とし、移動障害物が複数個存在する場合についてシミュレーションを行い、本手法の検討を行った。

まず、移動障害物が2個存在する場合におけるシミュレーションを行った。画像処理のようすを Figure 5.15 に示す。Figure 5.15 (1) は移動物体の連続画像を表し、Figure 5.15 (2) はそのうちの一枚の画像を表している。Figure 5.15 (3) は濃淡画像を表し、Figure 5.15 (4) は仮想注目点を決定するためにエッジ探索を行っているところである。ここでは移動ロボットに対して移動障害物の相対速度が比較的小さい場合と、比較的大きい場合についてのシミュレーションを行い、その結果を Figure 5.16、Figure 5.17 にそれぞれ示す。初期設定として  $P_r = [0, 0]^T$  ,  $V_r = [0, 30]^T$  ,  $P_G = [0, 200]^T$  , サンプルングタイム  $ST = 0.5[\text{sec}]$  である。障害物に関しては Figure 5.16 では、 $P_{o1} = [31, 107]^T$  ,  $V_{o1} = [-7, -12]^T$  ,  $P_{o2} = [-39, 138]^T$  ,  $V_{o2} = [+5, -4]^T$  とし、Figure 5.17 では、 $P_{o1} = [31, 107]^T$  ,  $V_{o1} = [-20, -20]^T$  ,  $P_{o2} = [-39, 138]^T$  ,  $V_{o2} = [+15, -5]^T$  とした。これらのシミュレーション結果により、余裕のあるときは障害物の手前を横切り、接近速度が大きくて危険なときは障害物の後方を回避していることがわかる。

次に、移動環境に3個以上の障害物が移動している場合のシミュレーション結果を Figure 5.18、Figure 5.19、Figure 5.20 にそれぞれ示す。いずれの場合においても、移動ロボットは移動障害物をうまく回避してゴールに到達していることがわかる。

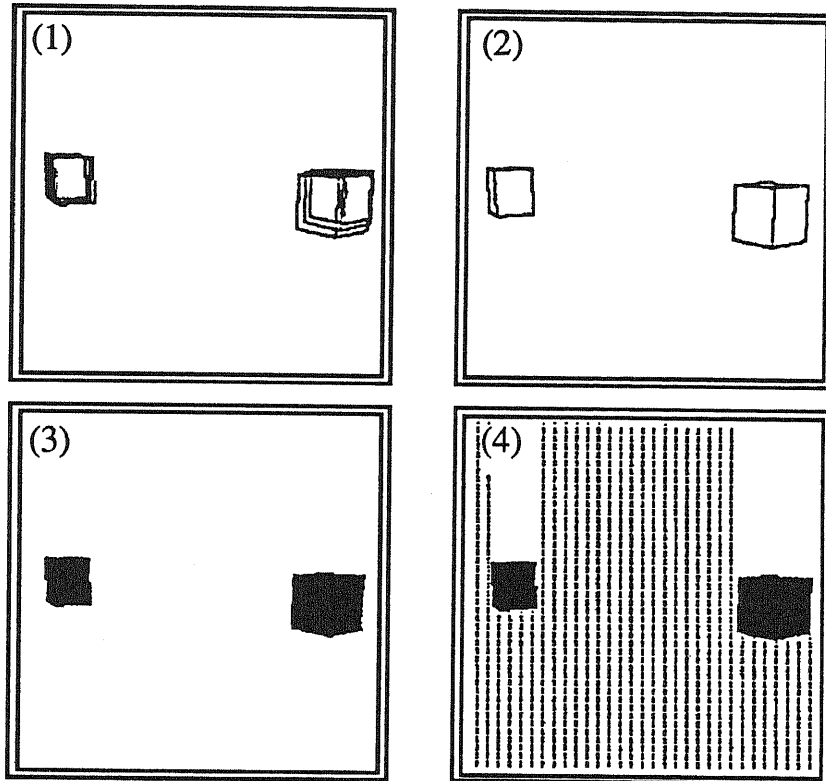


Figure 5.15 Simulation results 1-(1)  
[image processing]

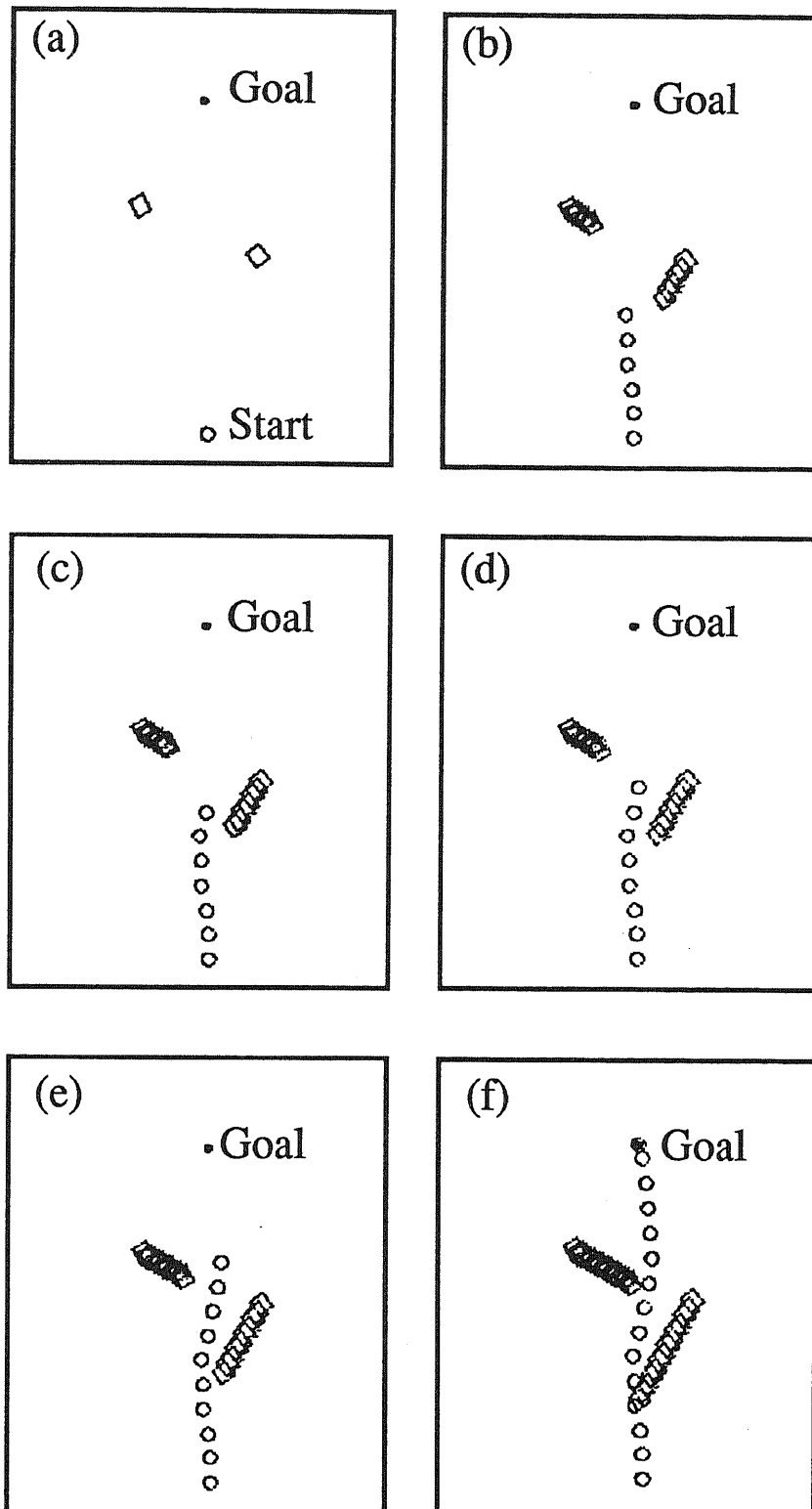


Figure 5.16 Simulation result 1-(2)

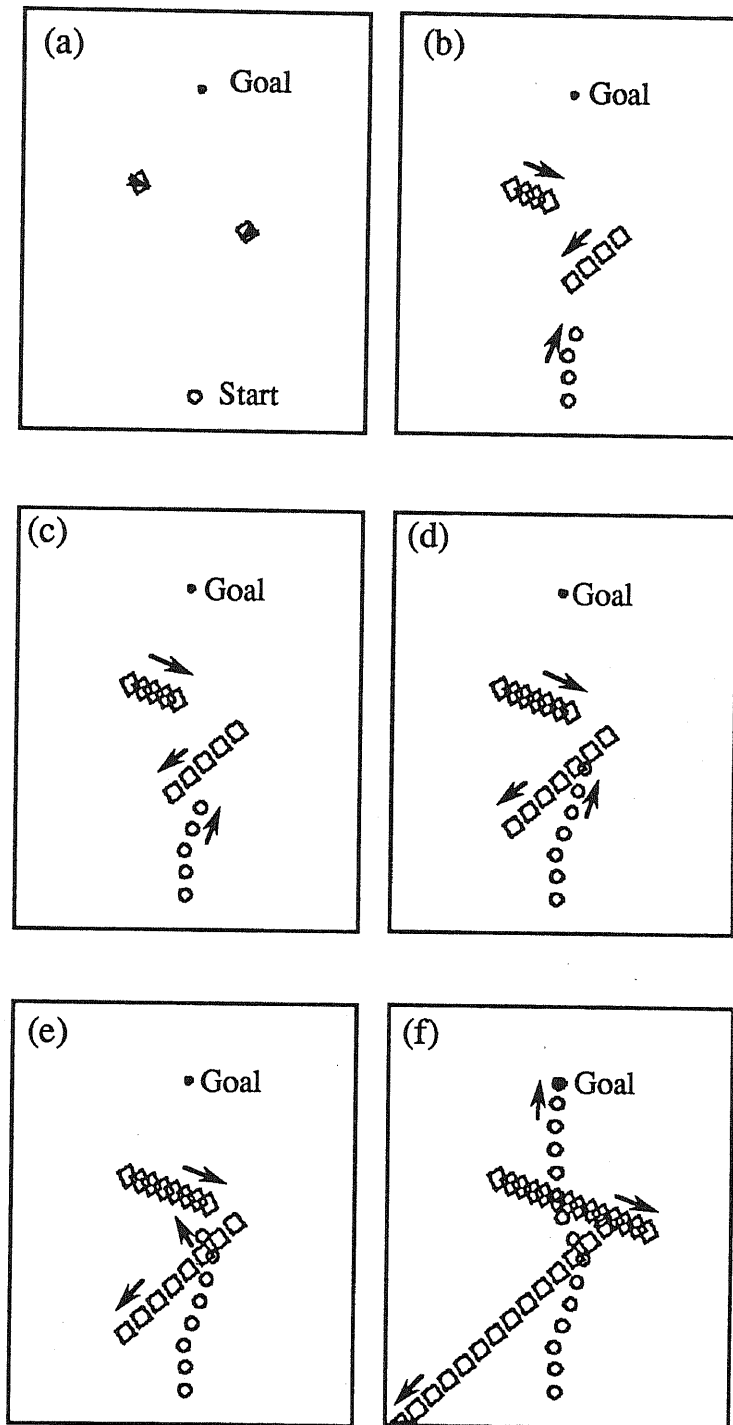


Figure 5.17 Simulation result 1-(3)

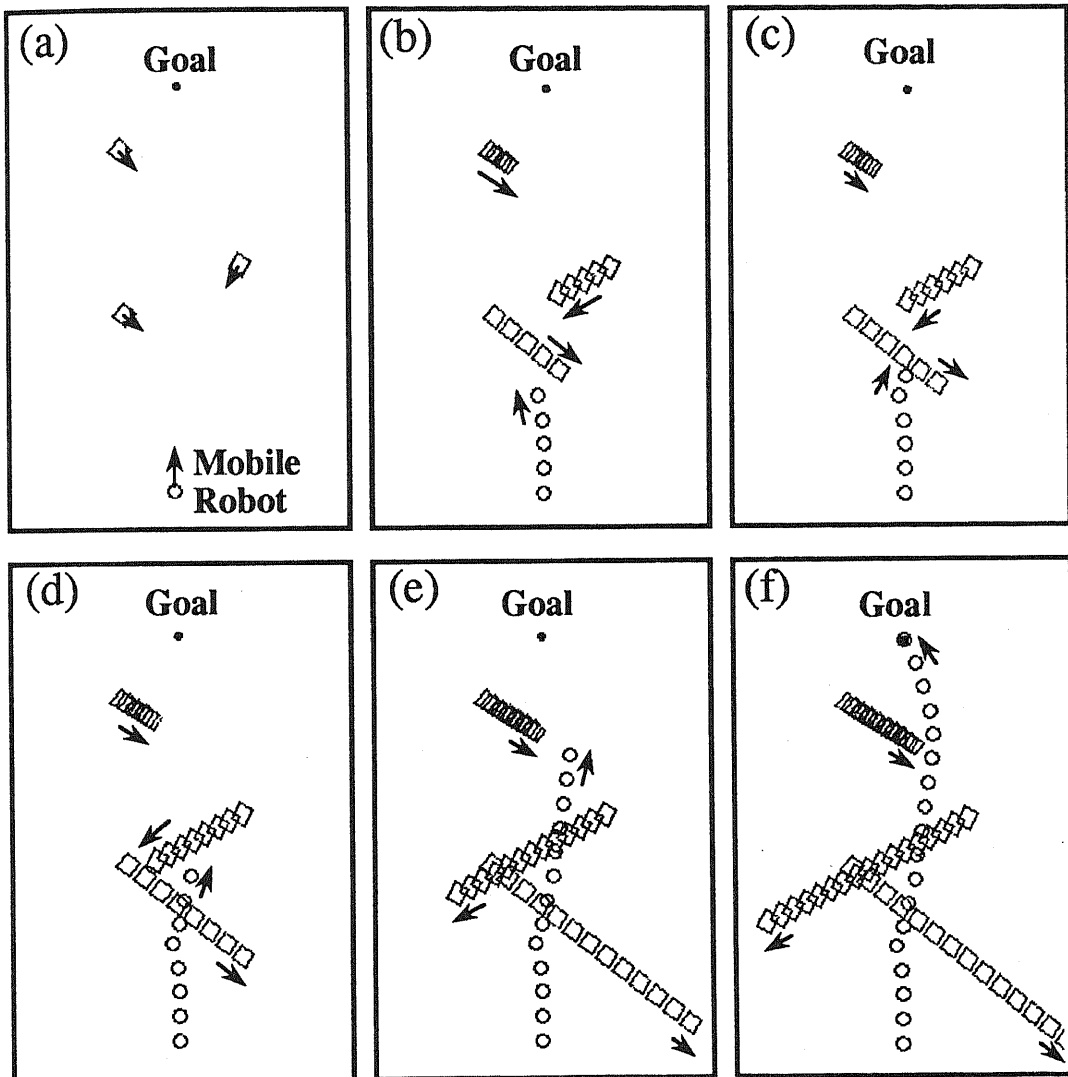


Figure 5.18 Simulation result 2

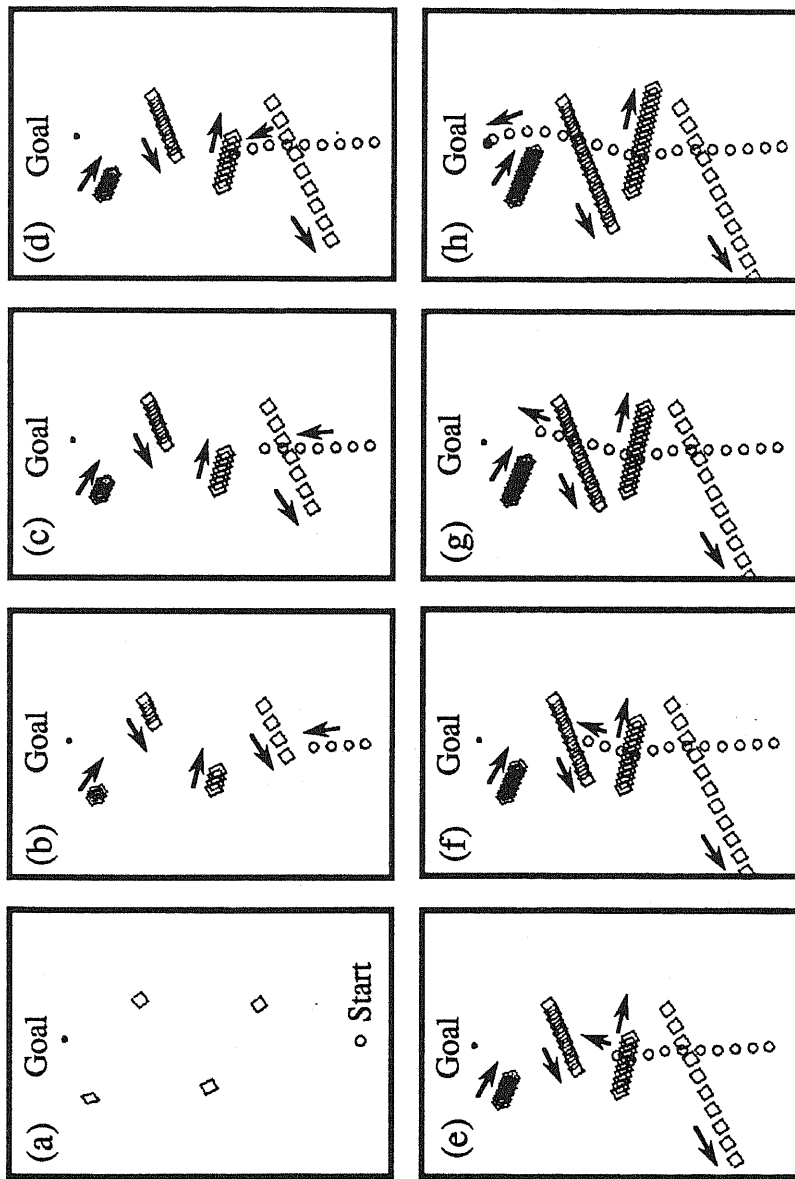


Figure 5.19 Simulation result 3

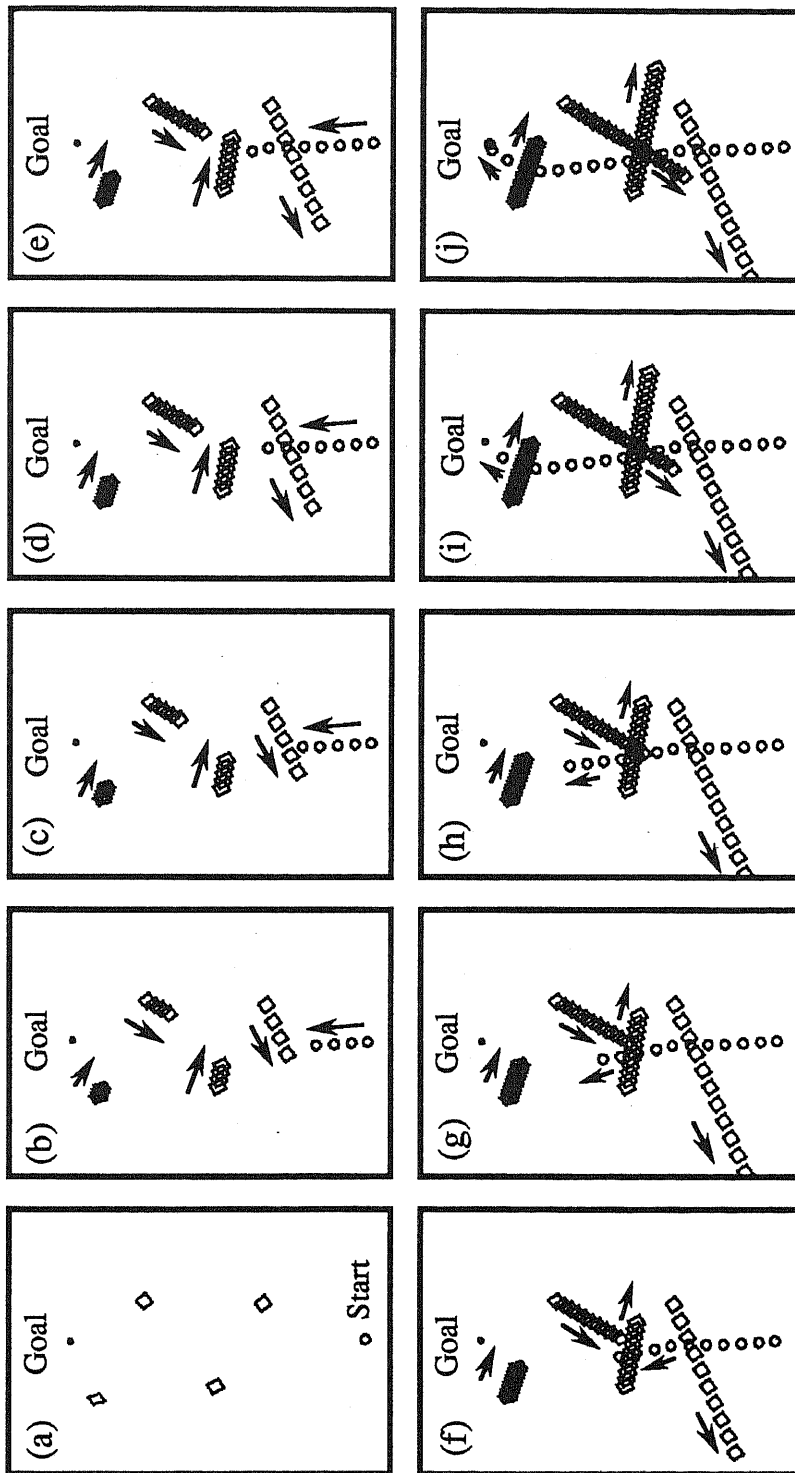


Figure 5.20 Simulation result 4



## 5.7 画像雑音に対する考察

本手法では、画像情報から障害物の運動情報を検出している。画像情報は3次元空間を2次元平面へ連続値から離散値に写像しており、その分解能によって画像解析した結果に大きな影響を与える。また、障害物の位置情報を得るために、物体の特徴点として画像からエッジと頂点を検出しているが実画像において雑音の影響は無視できず、誤差が生じる。さらに本手法では速度情報は位置情報の差分によって求めているので、得られた運動情報にはかなりのあいまいさが含まれることになる。ここでは、分解能の問題と画像雑音による影響についてシミュレーション結果をまじえながら考察する。

### 5.7.1 分解能に関する検討

視覚センサ（カメラ）によって対象物体の運動情報を検出するために、物体を可能な限り「良く」みることが重要であり、カメラの配置が重要な問題となる。また、正面に近いが距離が遠い場合と斜めからであるが近い場合とカメラの配置としてどちらがよいかなどの問題もある。さて、単眼視によって床面上での物体の位置を算定する場合、画像上での特徴点の位置座標から計算によってもとめるため、カメラの分解能の影響が位置情報の精度に影響を与える。したがって、画像の解像度およびカメラの配置が重要となる。

カメラの分解能を次のように考える。床面上の平面を考え、その平面をカメラで観測するものとする。この平面に対して、逆にカメラの撮像面上の各画素を投影する。一画素に相当する投影面積の拡大率の逆数が分解能の目安となる。すなわち、一画素の見込む面積が小さいほど高い分解能を得ていることになる。カメラの配置の評価としては、画像全体での分解能の平均（平均分解能）と画像内で分解能が最低になる点での分解能（最低分解能）などが画像全体での分解能と考えられている [出口 90]。

いま、平面上の注視点（画像の中心に対応する点）までの距離を  $D$ 、平面とカメラのなす角度を  $\beta$  とする。カメラの焦点距離  $F = 1$  とし、撮像面は一辺が  $2l$  の正方形でそこに  $N$  個の画素セルが格子状に配列されているものとする。平均分解能  $R_{av}$  は撮像面の面積とカメラが見込む領域の比から、最低分解能  $R_l$  はカメラからいちばん遠いところの一画素の見込む面積から得られる。

$$R_{av} = \frac{\{\cos^2 \beta_1 - (l \sin \beta_1)^2\}^2}{4l^2 D^2 \cos \beta_1} \cdot N \quad (5.14)$$

$$R_{is} = (1 - l \tan \beta_1)^3 \cdot \frac{\cos \beta_1}{D^2} \quad (5.15)$$

$$\beta_1 = \frac{\pi}{2} - \beta \quad (5.16)$$

となる。上式を用いて分解能に基づくカメラの配置問題を議論することが出来る。たとえば、 $D$  の値を一定にし、 $\beta_1$  の値を大きくしていくと高い分解能が得られる。これは、カメラを傾けて配置することでより高い分解能が得られる。また、最低分解能の要求を満たすカメラの配置領域なども求めることが出来る。

Figure 5.21 にカメラのパラメータをいろいろと変えたときのカメラからみた格子平面の画像を示す。このように、移動ロボットの動く環境に応じて、すなわちどの程度の精度の位置・速度情報が必要かによってカメラの配置および焦点距離を調節する必要がある。また、画像上での一画素の重み（実際に相当する距離）は、式(5.4)、式5.5を画素変数において微分し計算することにより求められる。第(5.6)節におけるシミュレーション (Figure 5.17) において調べてみると、近いところでは、進行方向に対しては一画素当り約1、横方向に対しては約0.2の距離重みである。一方、遠いところでは、進行方向に対しては一画素当り約5、横方向に対しては0.5の距離重みである。ファジィ推論をおこなうとき、これらの分解能を考慮してメンバシップ関数を設定することが必要となる。

### 5.7.2 画像ノイズに対する検討

本手法の画像雑音に対する影響を調べるために簡単なシミュレーションを行った。環境に関する初期設定は、第5.6節におけるシミュレーション (Figure 5.17) と同じにする。画像データに解像度の約2%のノイズと約5%のノイズをいれ、そのシミュレーション結果を Figure 5.22 と Figure 5.23 にそれぞれ示す。Figure 5.22 より、解像度の2%のノイズに対してはそれほど影響をうけないが、5%を越えてくると物体の運動推定のあいまいさが吸収できずに行動決定に悪影響を及ぼす。したがって、ノイズの少ない環境においては、仮想注目点設定によるシンプルな方法でよいが、屋外のようにノイズの多いところでは、オプティカルフローやカルマンフィルタなどを用

いて信号処理のレベルから運動情報を推定する必要があると思われる。あるいは、他のセンサを併用して精度をあげることも考えられる。ただし、センサから得られる運動情報によってメンバシップ関数などのチューニングは必要になるが、本手法の枠組みに関しては問題なく適用可能と思われる。

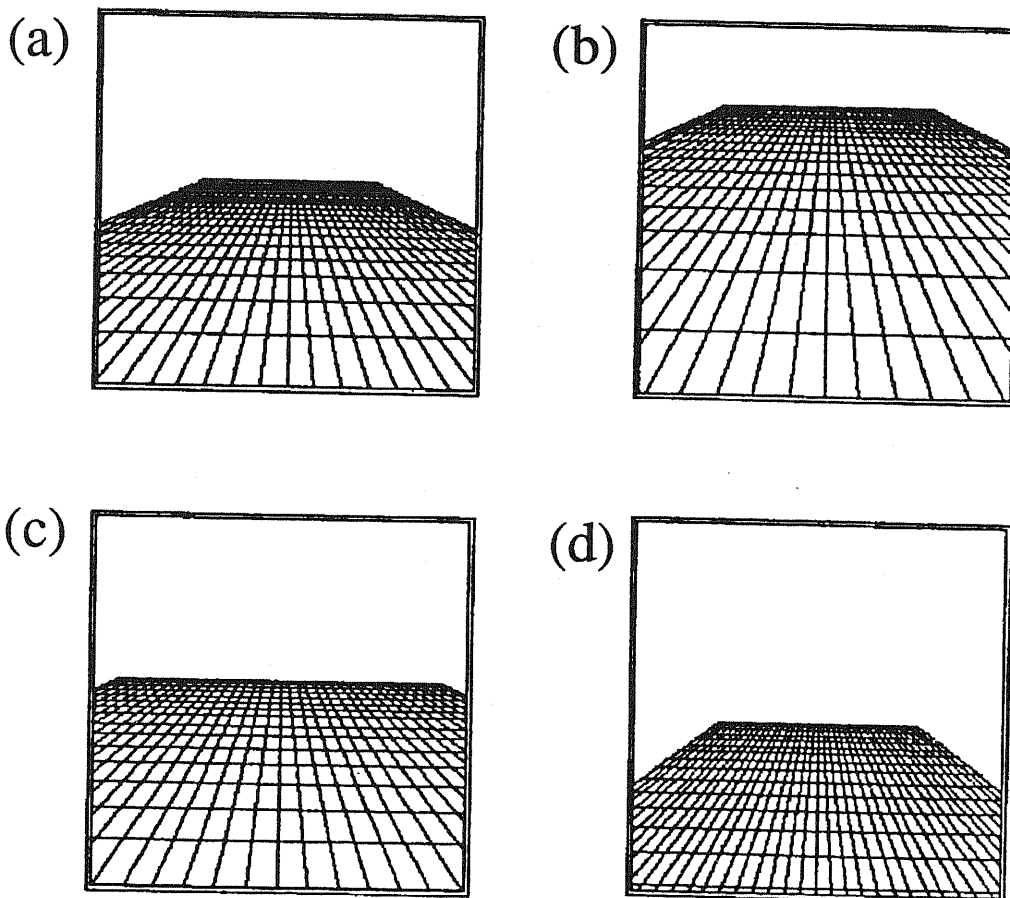


Figure 5.21 Image data

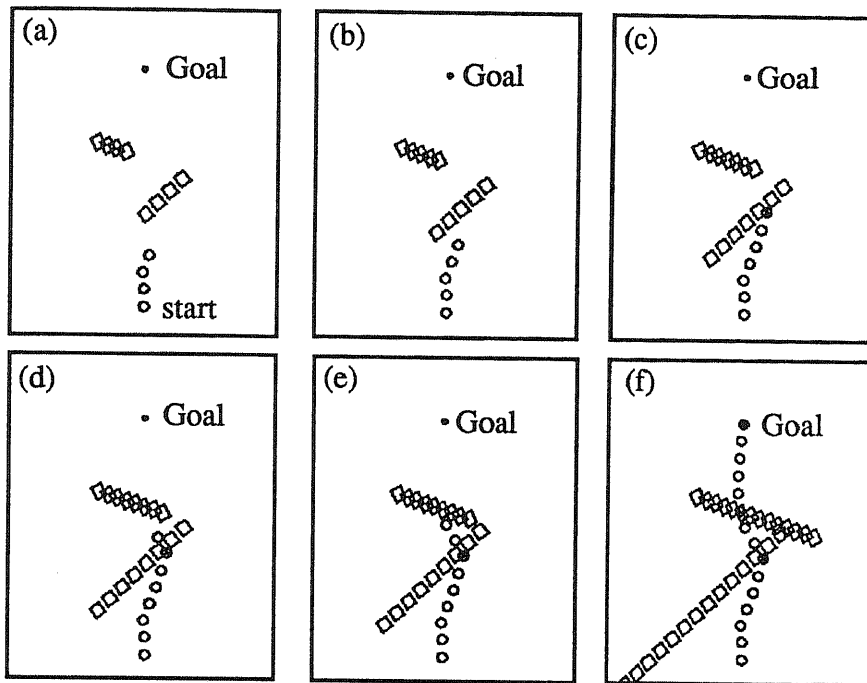


Figure 5.22 Simulation result with image noise (2%)

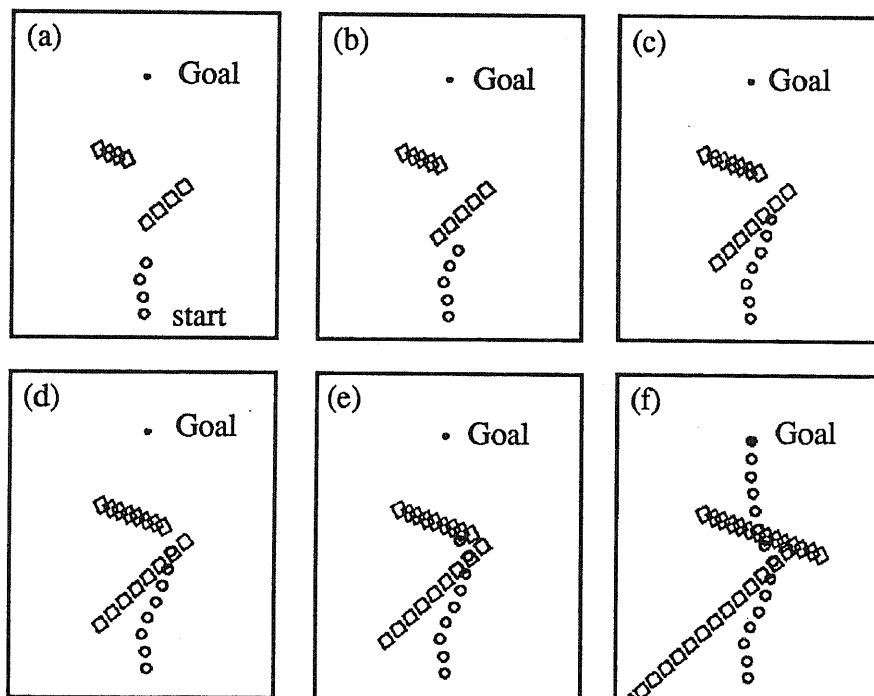


Figure 5.23 Simulation result with image noise (5%)

### 5.8 他の手法との比較検討

ここでは、本手法とポテンシャル法との比較を行なう。シミュレーション結果を Figure 5.24 に示す。Figure 5.24(a) はポテンシャル法による障害物回避、Figure 5.24(b) は本手法による障害物回避のシミュレーション結果である。これより、本手法によって知的な回避行動が形成されていることがわかる。

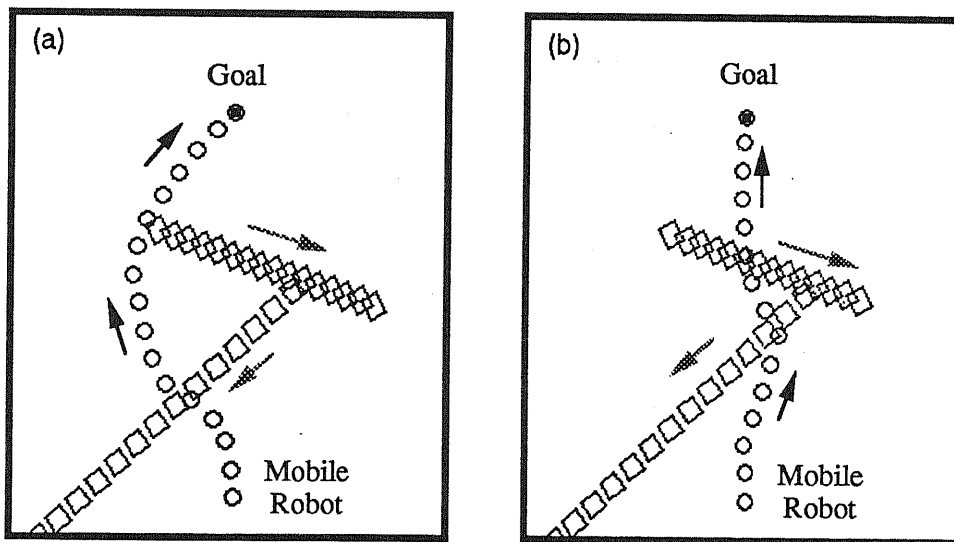


Figure 5.24 Simulation results

## 5.9 まとめ

移動障害物を回避するために、まず視覚センサを用いて移動物体のおおまかな運動状態を推定し、ファジィ推論によって評価された危険度をもとに回避行動を形成する方法について述べた。本手法を用いることにより人間に近い状況判断能力をロボットに与え、柔軟な回避行動を実現することができる。また、本手法を用いることにより動的な障害物に対してもフレキシブルに対応ができるだけでなく、ルールの変更による回避性能の変化も実現できる。

本手法の長所の1つは、メンバシップ関数およびルールを用いて表現することにより回避戦略が容易に記述・変更ができる点にある。従来法のようにアルゴリズムをシーケンスプログラムですべて記述すると戦略の変更はプログラムレベルでないと修正できないが、本手法のようにファジィ推論およびプロダクションシステムを用いて表現することによりアルゴリズムのメンテナンス性が向上すると思われる。

障害物回避アルゴリズムを決定する際に重要な問題は、メンバシップ関数およびルールの決定にあるが、環境に応じて人間が試行錯誤的に与えているのが現状である。今後の課題としては、人間が持っている回避行動に関するスキルを自動獲得（学習）し、メンバシップ関数の自動チューニングおよびルールの自動変更・追加などの機能を実現することが考えられる。

## 参考文献

- [Bhatt87] R.Bhatt, D.Gaw, A.Meystel, "A Real Guidance System for an Autonomous Vehicle," Proc. of IEEE Int. Conf. on Robotics and Automation, Vol.3, pp.1785-1791 (1987).
- [Brooks85] A.R.Brooks, "A Robust Layered Control System for a Mobile Robot," IEEE Journal of Robotics and Automation, Vol.RA-1, No.1 (1985).
- [Crowly85] J.L.Crowly, "Navigation for an Intelligent Mobile Robot," IEEE Journal of Robotics and Automation, Vol.RA-2, No.1 (1985).
- [出口 90] 出口, 高三, 牧野, "コンピュータビジョンのためのカメラ配置," 第29回計測自動制御学会学術講演会予稿集, pp.169-170 (1990).
- [Freund88] E.Freund and H.Hoyer, "Real-Time Path-Finding in Multi-Robot System Including Obstacle Avoidance," Int. Journal of Robotics Research, Vol.7, No.1, pp.42-47 (1988).
- [Fujimura89] K.Fujimura and H.Samet, "A Hierarchical Strategy for Path Planning Among Moving Obstacles," IEEE Tran. on Robotics and Automation, Vol.5, No.1 (1989).
- [藤原 90] 藤原, 米澤, 津村, "画像情報を利用した移動体の位置姿勢計測法," システム制御情報学会論文誌, Vol.3, No.4, pp.110-117 (1990).
- [Hashimoto88] H.Hashimoto, T.Kubota, F.Harashima, "Path Planning of Mobile Robot Based on Visual Information," Proc. of the USA-JAPAN Symp. on Flexible Automation. pp.637-644 (1988).
- [Isik86] C.Isik, A.Meystel, "Decision Making at a Level of a Hierarchical Control for Unmanned Robot," Proc. of IEEE Int. Conf. on Robotics and Automation, pp.1772-1778 (1986).
- [Kant88] K.Kant, and S.Zucker, "Planning Collision-free Trajectories in Time-varying Environments: a Two-Level Hierarchy," Proc. of IEEE Int. Conf. on Robotics and Automation, pp.1644-1649 (1988).
- [Khatib86] O.Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," Int. Journal of Robotics Research, Vol.5, No.1, pp.90-98 (1986).

- [Koch85] E.Koch, C.Yeh, G.Hillel, A.Meystel, C.Isik, "Simulation of Path Planning for a System with Vision and Map Updating," Proc. of IEEE Int. Conf. on Robotics and Automation, pp.146-160 (1985).
- [今野 89] 今野, 武野, "移動ロボットの衝突回避に関する基礎研究その13," 第7回日本ロボット学会学術講演会, No.1106 (1989).
- [久保田 87] 久保田, 橋本, 原島, "階層的プランニングによる経路探索(実験的検証)," 第5回日本ロボット学会学術講演会予稿集, pp.349-352 (1987).
- [Kubota88] T.Kubota, H.Hashimoto, and F.Harashima, "Path Searching of Mobile Robot Based on Cooperation of Sensors," Proc. of IEEE Int. Workshop on Intelligent Robots and Systems, pp.569-574 (1988).
- [久保田 89] 久保田, 橋本, 原島, "視覚情報を用いた移動ロボットの障害物回避(移動障害物に対する回避行動の形成)," 第7回日本ロボット学会学術講演会, No.1103 (1989).
- [Kubota90] T.Kubota, H.Hashimoto, "A Strategy for Collision Avoidance among Moving Obstacles for a Mobile Robot", Proc. of 11th World Congress of International Federation of Automatic Control, Vol.9, pp.103-108 (1990).
- [武野 86] 武野, 柿倉, "移動ロボットの移動障害物に対する衝突回避問題について," 日本ロボット学会誌, Vol.4, No.5, pp.33-37 (1986).
- [前田 88] 前田, 竹垣, "ファジィ推論を用いた移動ロボットの動的障害物回避制御," 日本ロボット学会誌, Vol.6, No.6, pp.50-54 (1988).
- [Rives85] P.Rives and L.Marce, "Use of Moving Vision Sensors in Robotics Application to an Obstacle Avoidance Task," Proc. of IJCAI 85, pp.215-223 (1985).
- [斉藤 88] 斉藤, 津村, "移動ロボットシステムにおける衝突回避," 第4回知能移動ロボットシンポジウム講演論文集, pp.85-90 (1988).
- [斉藤 90] 斉藤, 津村, "移動ロボット間における衝突回避," システム制御情報学会論文誌, Vol.3, No.1, pp.1-7 (1990).
- [菅野] 菅野, "ファジィ制御," 日刊工業新聞社.
- [鈴木 90] 鈴木, 有本, "時変環境における点状の移動ロボットのペイント手続きを利用した再帰的軌道計画法," 日本ロボット学会誌, Vol.8, No.5, pp.9-18 (1988).



- [Tsuji85] S.Tsuji, Y.Yagi, and M.Asada, "Dynamic Scene Analysis for a Mobile Robot in a Man-made Environment," Proc. of IEEE Int. Conf. on Robotics and Automation. pp.850-855 (1985).
- [八木 86] 八木, 浅田, 辻, "移動カメラで撮像した環境内の運動物体の検出," 電子通信学会論文誌, Vol.J69-D, No.5, pp.794-801 (1986).

## Chapter 6

### 結 論

本研究のまとめと今後の課題について述べる。

本研究では、実時間処理を念頭に、移動ロボットシステムにおいてセンサから得られる情報と運動系の制御情報の整理・統合を行い、センサ情報処理系と行動制御系との有機的な結合を実現することによって、センサベースロボットコントロールシステムを構成することを大目的とし、次の3つのフェーズに注目した。

#### Phase\_1 移動ロボットの行動計画

#### Phase\_2 移動ロボットの制御

#### Phase\_3 移動ロボットの行動決定

本研究では、[Phase\_1] から [Phase\_2]、さらに [Phase\_3] に進むに従って移動ロボットが取り扱う環境が複雑になる方法をとっている。まず、[Phase\_1] では移動ロボットの経路探索問題を取り扱い、地図情報とセンサ情報を有効利用する階層型パスプランニング法を確立した。特に視覚情報に基づくローカルパスプランニング手法はアルゴリズムをシンプルにすることによりリアルタイム処理が可能となっている。[Phase\_2] では、[Phase\_1] の経路計画に基づいて目標までロボットを誘導する、移動ロボットのナビゲーション問題を取り扱った。環境が複雑になると単眼視による環境理解のあいまいさ故、他のセンサからの情報を必要とする。[Phase\_2] では視覚センサと超音波センサを用いて、これをセンサ情報融合手法として確立した。[Phase\_3] では、さらに複雑な環境を取り扱った。移動ロボットの動的環境下における衝突回避問題を設定し、ファジィ推論およびプロダクションルールによって柔軟な回避行動を実現した。

以下に、第1章をのぞく各章で得られた知見と問題点をまとめる。

第2章「ロボットとセンサ情報」では、現在実用化されている産業用ロボットの問題点を挙げ、ロボットの知能化の必要性を論じた。次に移動ロボットの重要な課題である経路探索について、環境に関する知識の度合によって、3つのケースに分類し、すでに提案されている手法の検討を行った。また、ビジョンベースシステムに関して、ロボットにおける視覚情報の役割について述べ、視覚情報の大域性に着目したロボットの行動制御手法の確立を強調した。さらに、センサ融合システムに関して、研究の現状と課題について概観するとともに、センサフュージョンの考え方について検討を加えた。

第3章「視覚情報を用いた移動ロボットの経路探索」では、あいまいな環境において有効な経路探索方法として、階層的プランニング方法を提案した。これはスタートからゴールまでの経路をサブゴールマップの概念を用いて決定するグローバルパスプランニングと、障害物回避などの実際に移動しながらロボットの環境に応じて行うローカルパスプランニングとからなる。この方法は環境地図とセンサ情報を有効に利用できるという特徴をもつ。また、ローカルプランニングにおいて、単眼視による環境観測をおこなない、画像データから回避ポイントの自動生成を試み、シミュレーション及び実験によってその有効性を確認した。また、複数障害物識別のための画像探索法、ナビゲーション法を考案した。ローカルパスプランニング手法は、視覚センサを用いて未知の障害物を回避するためのポイントを逐次選択するアルゴリズムである。本手法の特徴は、画像情報の大域性を利用し、リアルタイム処理が可能な程度にシンプルな点であり、実用化が可能である。今後の課題としては、グローバルプランニングにおいて画像地図データからクワッドツリー手法を利用してサブゴールを生成する可能性を述べたが、人間が与える大域的な地図の新しい表現方法、及びどのような評価関数で経路を決めるかなどの問題がある。

第4章「センサの協調による移動ロボットのナビゲーション」では、まず単眼視による環境観測を行い、視覚情報を用いた移動ロボットのナビゲーションを検討した。環境が複雑になるにつれて単眼視による環境観測にはあいまいさが増大する。そこで、このあいまいさを減少させるために、超音波センサ等の近接センサを同時に用いる。ここでは、ロボットが必要な情報を信頼性高く獲得するために異種センサの併用・協調を行い、そのセンサ情報処理系をセンサ情報融合システムとして位置づける。この融合システムを移動ロボットのコントロールという観点から検討を行い、複合的処理、統合的処理、融合的処理の3つの処理形態を提案し、実際に視覚センサ情報とレンジセンサ情報の融合方法を提案した。本手法の有効性をシミュレーション及び実験によって確認した。提案した融合方法では、視覚情報を距離情報に直して融合しており、大域的な情報である視覚情報を十分に利用しているとはいえ、異種センサ情報の融合に関する新しい手法が望まれるところである。しかしながら、センサフュージョンは、まだ新しい研究分野であり、考え方自体明確に整理されておらず、また現状では多種類のセンサを組み合わせることはそれほど簡単ではない。本手法では、移動ロボットの制御情報を異種センサである視覚センサとレンジセンサから生成しており、一種の融合とみなすことができ、この分野の研究を一步前進させたものと考えられる。

第5章「動的環境における移動ロボットの行動制御」では、移動障害物を回避するために、まず視覚センサを用いて移動物体のおおよその運動を推定した。本論文では、リアルタイム処理を念頭に、障害物回避のため仮想注目点の考えを導入した。得られた情報にはあいまいさがあり、移動体の運動状態をファジィで表現する。ファジィ推論によって評価された危険度をもとに回避行動を形成する方法を提案した。本手法の有効性はシミュレーションによって確認した。本手法を用いることにより人間に近い状況判断能力をロボットに与え、柔軟な回避行動を実現することができる。本手法の利点は、メンバシップ関数およびルールを用いて表現することにより回避戦略が容易に記述・変更ができる点にある。しかしながら、メンバシップ関数などの決定は人間の手に委ねられており、今後の課題として、人間が持っている回避行動に関するスキルを自動獲得（学習）し、メンバシップ関数の自動チューニングおよびルールの自動変更・追加などの機能を実現することが必要である。

自律型移動ロボットの研究は、従来のロボットがもつ問題をほとんどすべて網羅するとともに、新しい数多くの問題提起と示唆を与えてくれる領域である。自律型移動ロボットに関する研究はまだ立ち上がったばかりであり、解決すべき課題が山積している。本研究は其中で重要な課題のいくつかに対し、センサ情報処理系と行動制御系との有機的な結合を実現するという観点からのアプローチを行った。現実世界を対象とするロボットは、センサデータの取り扱いが大きな問題であり、ロボットシステムにおけるセンサ処理系が重要な課題となっている。本研究でもこのセンサ情報の取り扱いをロボットシステム全系のなかで位置づけ、そのデータ構造を上位レベルの処理に適したものにすることを念頭においている。また、センサ情報融合の考え方は、知能ロボットに限らず、自律的な機能を必要とする自動化技術において、システムの感覚系の構造を左右する重要な問題であり、この意味でも本研究の意義は大きいと思われる。この分野に関する研究が今後共進歩し続けることを期待してやまない。

## Appendix A

### マニピュレータのビジュアル制御

マニピュレータの先端にカメラを取り付け、視覚情報を用いて先端を目標の位置及び姿勢に移動させる方法について述べる。複雑な幾何学的計算を行わずに、並列性と柔軟性に富んだニューラルネットワークを用いることによって、画像データからマニピュレータの各関節の制御入力情報を直接得る方法を提案する。

## A.1 はじめに

ロボットが知的な動作をするためには、広範囲な環境に対して適応する能力が必要である。この能力を付与するために、視覚を有するロボットの研究が盛んに行われている [佐藤 87]。従来のビジュアルフィードバック制御では、視覚情報を画像解析したうえで、ロボットを制御する状態量に変換してフィードバックする方式が主に用いられており、演算処理に時間がかかるなどの問題がある。これに対して最近では、視覚系とロボット制御系の密な結合を実現するために、視覚特徴量を直接マニピュレータの制御に結び付ける研究が行われている [Sanderson85][Anderson85]。これは、画像上の物体の面積や線分の長さなどの画像特徴量を用いてマニピュレータを直接的に制御し、制御系のダイナミックな特性を向上させようというものである。このようなイメージ情報に基づいた制御法では、画像解釈を行わないので計算機の負担が軽くなるという利点がある。

一方、柔軟なシステムの構築をめざして、知識処理、パターン認識、制御などさまざまな分野で、ニューラルネットワークの研究、およびその応用が活発に行われている [甘利 72][麻生 88]。特に、ニューラルネットワークは並列分散処理機能や学習機能を有しているため、画像処理およびロボット制御の分野において大きな期待が寄せられている [Kawato87][Guez88][Martinetz90]。最近、センサ情報に基づいてコントロール情報を生成するために、ニューラルネットワークをロボットの制御に適用する研究が精力的に進められている。両眼視から得られた画像データを基にニューラルモデルを用いて制御を行うハンドアイシステムに関する研究 [田中 88][Kuperstein89][Krose90] や小脳の神経回路モデル (CMAC) をマニピュレータ制御に適用する研究 [Albus75][Miller87,89][Kano90] などが提案されている。

そこで、本章では視覚情報を直接利用することによって、マニピュレータの先端を目標位置及び目標姿勢に移動させるビジュアルコントロール法について述べる。画像特徴量としては、対象物の特徴点4点に着目する。その4点があらかじめ与えられた目標値に一致するようにマニピュレータへの制御入力を生成する。ここでは、幾何学的な座標変換を行わずに、並列性と柔軟性に富んだニューラルネットワークを用いることによって、画像データとマニピュレータの関節角情報との有機的な結合を行う。また、6自由度マニピュレータを対象としてシミュレーションを行い、提案する制御方式の有効性について検討を行う。また、実験システム (5自由度マニピュレータ) を用いて実験的検討を行う。

## A.2 問題設定

### A.2.1 ピック&プレイス

マニピュレータの代表的な応用例として、ピック&プレイス作業が考えられる。この作業は、Figure A.1 に示すように次の3つの手順からなる。

1. マニピュレータの先端を対象物に対して目標の位置・姿勢に移動させる。
2. 対象物をつかむ。
3. つかんだ対象物を他の場所に移動して置く。

ここでは1.の手順に注目する。1.の作業がうまく行われれば、続く2.および3.の作業は、比較的容易に行うことができる。即ち、視覚によって大まかな動作（大体の位置決め）を行い、マニピュレータのコンプライアンスによって小さな誤差を吸収することになる。このとき、対象物が所定の位置からずれた場合にも、対象物に対してマニピュレータの先端が与えられた目標の位置・姿勢に移動することが必要となる。

本研究では、Figure A.2 に示すようにロボットマニピュレータの先端に、あるオフセットをもってカメラが取り付けられており、カメラからの視覚情報をもとにマニピュレータの先端を目標の位置及び姿勢にコントロールするシステムを考える。一般に、画像情報からマニピュレータの各関節角への制御入力を得るには、次に示す4つのステップから成る座標変換が必要となる。

<ステップ1>：絶対座標系で表現される対象物をカメラ座標系に座標変換

<ステップ2>：対象物をカメラ座標系内の投影平面に透視変換

<ステップ3>：カメラ座標系からマニピュレータ先端の座標系への座標変換

<ステップ4>：マニピュレータ先端の座標系から各関節の座標系への座標変換



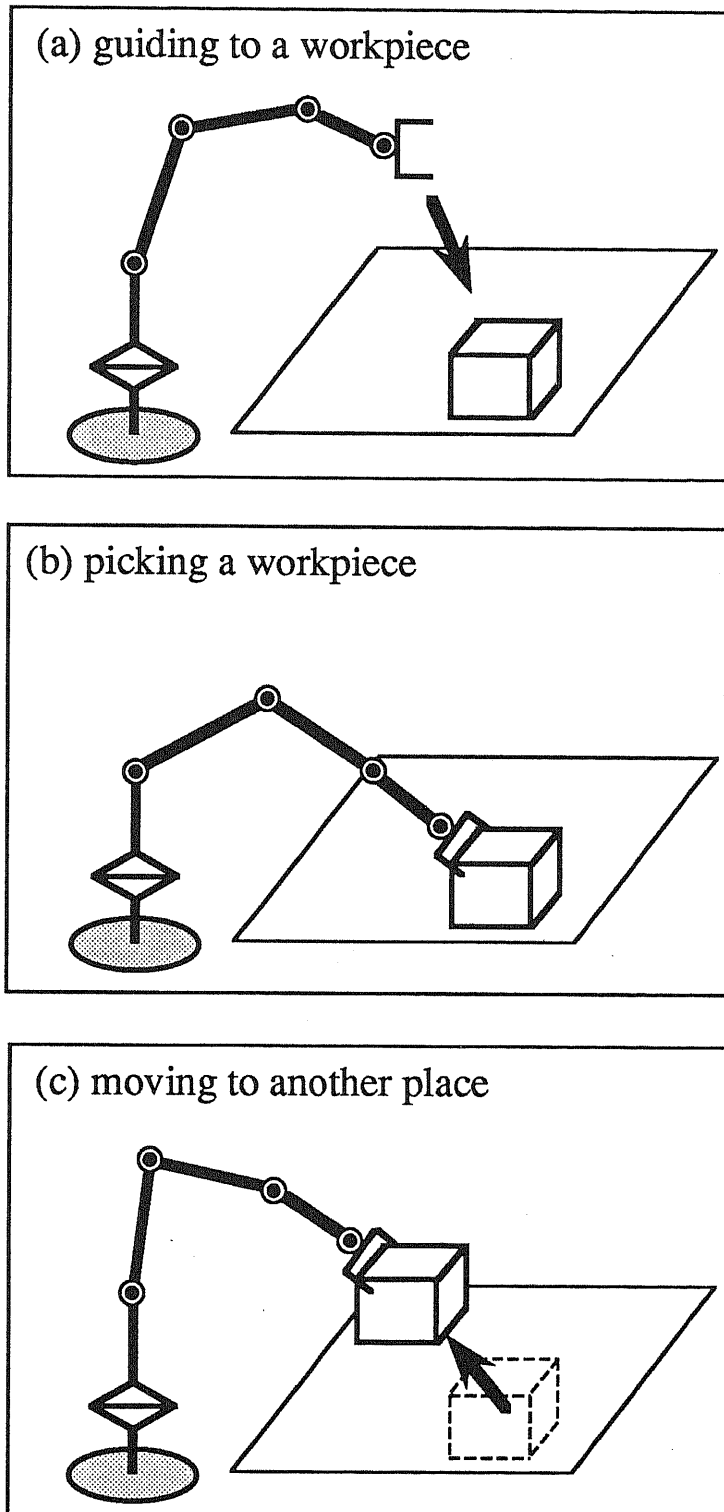


Figure A.1 Pick and place

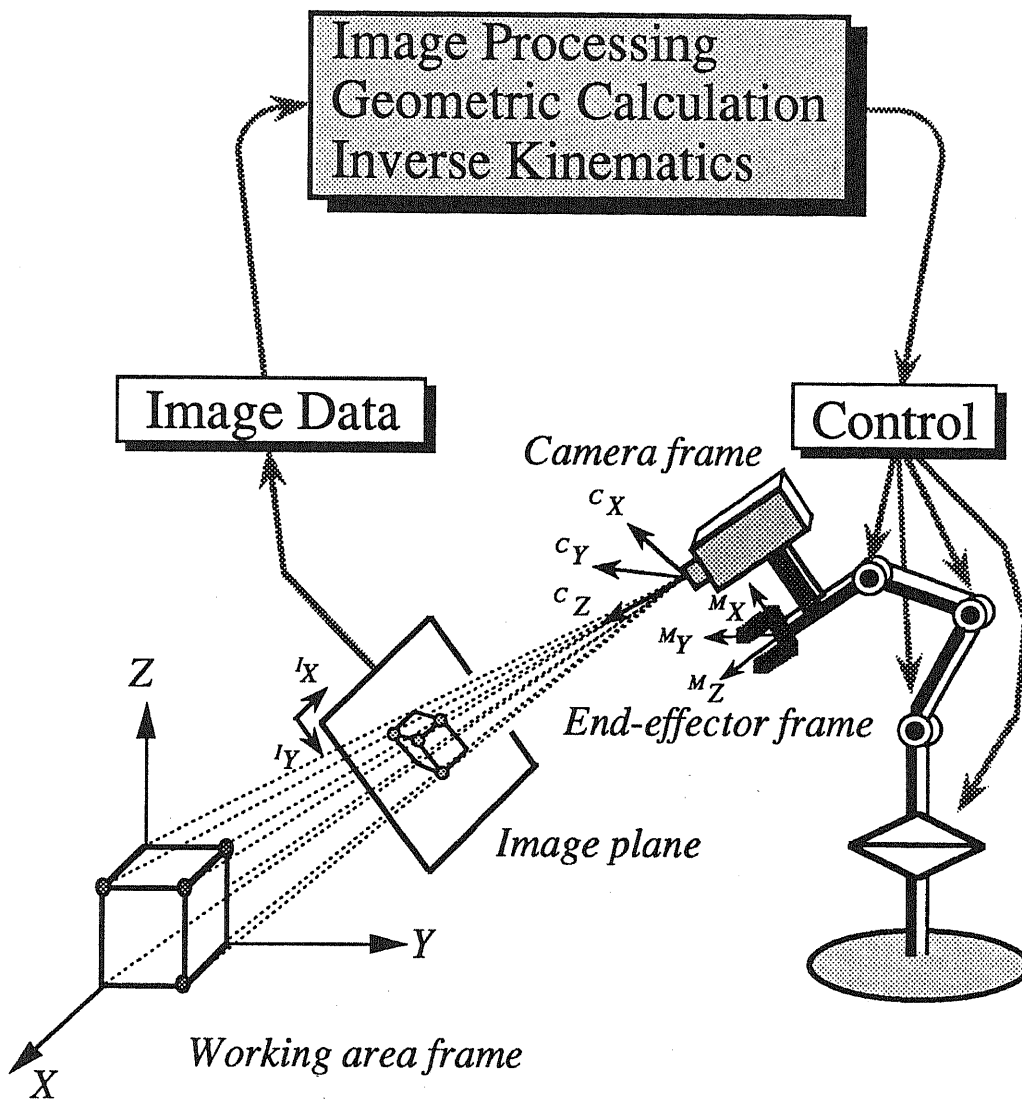


Figure A.2 Visual control

### A.2.2 座標変換

4つのステップの座標変換（Figure A.3 参照）を具体的に示すと次のように表せる。

#### <ステップ1>

絶対座標系でのカメラの位置を  $(X_c, Y_c, Z_c)$ 、オイラー角で表したカメラの姿勢を  $(\phi_c, \varphi_c, \psi_c)$ 、絶対座標系における対象物の特徴点の位置を  $(X_i^P, Y_i^P, Z_i^P)$ 、カメラ座標系における対象物の特徴点の位置を  $(X_i^C, Y_i^C, Z_i^C)$  とすると、ステップ1の座標変換は次に示す回転と平行移動によって表される。

$$\begin{bmatrix} X_i^C \\ Y_i^C \\ Z_i^C \end{bmatrix} = R^T \begin{bmatrix} X_i^P - X_c \\ Y_i^P - Y_c \\ Z_i^P - Z_c \end{bmatrix} \quad (\text{A.1})$$

ただし、

$$R = R\phi \cdot R\varphi \cdot R\psi \quad (\text{A.2})$$

$R\phi$  : X軸まわりに角度 $\phi$ 回転したときの回転行列

$R\varphi$  : Y軸まわりに角度 $\varphi$ 回転したときの回転行列

$R\psi$  : Z軸まわりに角度 $\psi$ 回転したときの回転行列

したがって、式(A.1)より、絶対座標系の  $(X_i^P, Y_i^P, Z_i^P)$  を平行移動と回転によってカメラ座標系の  $(X_i^C, Y_i^C, Z_i^C)$  に変換することができる。

#### <ステップ2>

カメラ座標系の座標値  $(X_i^C, Y_i^C, Z_i^C)$  をカメラの視点から焦点距離  $f$  だけ離れた所に、視線と垂直な投影平面を設定する。投影平面上の座標値を  $(X_i^I, Y_i^I)$  とすると、 $(X_i^I, Y_i^I)$  と  $(X_i^C, Y_i^C, Z_i^C)$  は次式によって表される。

$$\begin{bmatrix} h \cdot X_i^I \\ h \cdot Y_i^I \\ h \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_i^C \\ Y_i^C \\ Z_i^C \end{bmatrix} \quad (\text{A.3})$$

式(A.3)によって、3次元物体を2次元の画像平面に投影できる。

### <ステップ3>

ステップ3では、カメラ座標系  $(X_i^C, Y_i^C, Z_i^C)$  をオフセット分だけ平行、回転の座標変換を行い、マニピュレータ先端の座標系  $(X_i^M, Y_i^M, Z_i^M)$  に座標変換する。

### <ステップ4>

ステップ4では、マニピュレータの先端の位置  $(X, Y, Z)$ 、及び姿勢  $(\phi, \varphi, \psi)$  から各関節角度  $(\theta_j)$  を求めるために、逆キネマティクスを解く。

## A.2.3 画像情報からの位置・姿勢の決定

<ステップ3>、<ステップ4>の座標変換を行うためには、カメラの位置・姿勢を知ることが必要である。式(A.1)、式(A.2)、式(A.3)において、既知パラメータとしては、カメラの焦点距離  $f$ 、画像平面での特徴点の座標値  $(X_i^I, Y_i^I)$  ( $i = 1, 2, \dots, n$ ) である。未知パラメータとしては、カメラの位置  $(X_c, Y_c, Z_c)$  及び姿勢  $(\phi_c, \varphi_c, \psi_c)$  である。従って、式(A.1)、式(A.2)、式(A.3)からこの未知パラメータを決定するためには、もう一つ制約条件を付け加えることが必要となる。例えば、対象物の形状が既知であるという条件などを付け加えると、絶対座標系における特徴点間の長さ  $(L_{ij})$  を用いることによって対象物とカメラとの相対的な位置姿勢関係  $(X_c, Y_c, Z_c, \phi_c, \varphi_c, \psi_c)$  を求めることができる。この関係を次の式に示す。さらに、対象物に関する位置情報から絶対座標系におけるカメラの位置及び姿勢を求めることができる。

$$\begin{cases} X_i^I = f H_{xi} \cdot G_{xi}^{-1} \\ Y_i^I = f H_{yi} \cdot G_{yi}^{-1} \end{cases} \quad (\text{A.4})$$

$$\left\{ \begin{array}{l} H_{xi} = (C\phi C\varphi C\psi - S\phi S\psi)(X_i^P - X_c) \\ \quad + (S\phi C\varphi C\psi + C\phi S\psi)(Y_i^P - Y_c) + S\phi C\varphi(Z_i^P - Z_c) \\ G_{xi} = -C\varphi S\psi(X_i^P - X_c) + S\varphi S\psi(Y_i^P - Y_c) + C\varphi(Z_i^P - Z_c) \\ H_{yi} = (-C\phi C\varphi S\psi - S\phi C\psi)(X_i^P - X_c) \\ \quad + (-S\phi C\varphi S\psi + C\phi C\psi)(Y_i^P - Y_c) + S\varphi S\psi(Z_i^P - Z_c) \\ G_{yi} = C\phi C\varphi(X_i^P - X_c) + S\phi S\varphi(Y_i^P - Y_c) + C\varphi(Z_i^P - Z_c) \end{array} \right. \quad (\text{A.5})$$

$$\left\{ \begin{array}{l} L_{12} = \sqrt{(X_1^P - X_2^P)^2 + (Y_1^P - Y_2^P)^2 + (Z_1^P - Z_2^P)^2} \\ L_{13} = \sqrt{(X_1^P - X_3^P)^2 + (Y_1^P - Y_3^P)^2 + (Z_1^P - Z_3^P)^2} \\ L_{14} = \sqrt{(X_1^P - X_4^P)^2 + (Y_1^P - Y_4^P)^2 + (Z_1^P - Z_4^P)^2} \end{array} \right. \quad (\text{A.6})$$

ただし、

$$\left\{ \begin{array}{l} S\phi = \sin(\phi), \quad S\varphi = \sin(\varphi) \\ S\psi = \sin(\psi), \quad C\phi = \cos(\phi) \\ C\varphi = \cos(\varphi), \quad C\psi = \cos(\psi) \end{array} \right. \quad (\text{A.7})$$

しかしながら、実際に解を導くには、式(A.4)、式(A.5)、式(A.6)、(A.7)の非線形連立方程式を解かなければならず、大量の計算を要する。

一方、現在の多くの研究は、2次元の画像データから3次元形状の認識を行ったり、カメラと対象物との相対的な位置・姿勢を求めることにとどまっており、さらにそれからマニピュレータへの関節角度に変換する研究はあまりみられない。

そこで、本章では大量の演算を要する〈ステップ1〉～〈ステップ4〉の座標変換を行わずに、視覚情報を直接利用してマニピュレータの先端を目標の位置・姿勢に移動させる方法を提案する。まず、次節で本手法の背景となる、特徴点4点によるカメラ座標系に関する位置・姿勢関係について述べる。

## A.2.4 特徴点4点に基づくカメラ系の位置・姿勢関係

一般的に以下の命題が知られている。[Hung85][Feddem89]

## [命題A.1]

対象物の特徴点4点が視覚によって認識可能で、それら4点の幾何学的な位置関係が既知である場合、カメラのパラメータ、特徴点の画像上での座標（画素データ）、及び特徴点の幾何学的な知識を用いて、対象物に対するカメラ（手先）の位置及び姿勢が一意に定まる。

ここでは、特徴点4点が同一平面上にある場合について考える。[堀口85]いま、Figure A.4 に示すように3次元空間を直交座標系  $(X^c, Y^c, Z^c)$  とし、原点をレンズ中心におき、 $Z^c$  軸をレンズ光軸とする。平面  $S$  を考え、 $S$  上に座標系  $(P_L, P_M)$  を定め、 $L, M$  軸の3次元空間内の方向余弦をそれぞれ次のようにおく。

$$\begin{cases} P_L \text{の方向余弦} : L = (L_x, L_y, L_z) \\ P_M \text{の方向余弦} : M = (M_x, M_y, M_z) \\ \text{原点位置} : P = (P_x, P_y, P_z) \end{cases} \quad (\text{A.8})$$

平面  $S$  上の点  $q(\alpha, \beta)$  の3次元空間内の位置座標を  $Q(q_x, q_y, q_z)$  とすると、

$$Q = P + \alpha L + \beta M \quad (\text{A.9})$$

$$\begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} + \alpha \begin{bmatrix} L_x \\ L_y \\ L_z \end{bmatrix} + \beta \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} \quad (\text{A.10})$$

また、 $Z^c = f$ （焦点距離）の平面上に画像座標系  $(X^I, Y^I)$  を設定すると、

$$\begin{cases} X_i^I = f \frac{X_i^c}{Z_i^c} \\ Y_i^I = f \frac{Y_i^c}{Z_i^c} \end{cases} \quad (\text{A.11})$$

従って、平面  $S$  上の4点の相対位置関係を座標  $(\alpha_i, \beta_i)$  ( $i = 1, 2, 3, 4$ ) で表し、その頂点  $i$  の透視射影像  $(X_i^I, Y_i^I)$  は次のようになる。

$$X_i^I = f \frac{P_x + \alpha_i L_x + \beta_i M_x}{P_z + \alpha_i L_z + \beta_i M_z} = \frac{T_A + \alpha_i T_C + \beta_i T_D}{1 + \alpha_i T_G + \beta_i T_H} \quad (\text{A.12})$$

$$Y_i^I = f \frac{P_y + \alpha_i L_y + \beta_i M_y}{P_z + \alpha_i L_z + \beta_i M_z} = \frac{T_B + \alpha_i T_E + \beta_i T_F}{1 + \alpha_i T_G + \beta_i T_H} \quad (\text{A.13})$$

ただし

$$\begin{cases} T_A = \frac{fP_x}{P_z}, T_B = \frac{fP_y}{P_z}, T_C = \frac{fL_x}{P_z}, T_D = \frac{fM_x}{P_z} \\ T_E = \frac{fL_y}{P_z}, T_F = \frac{fM_y}{P_z}, T_G = \frac{L_z}{P_z}, T_H = \frac{M_z}{P_z} \end{cases} \quad (\text{A.14})$$

カメラと特徴点4点との関係は、次のようにして求める。まず、既知情報  $(\alpha_i, \beta_i)$  と  $(X_i^I, Y_i^I)$  からパラメータ  $T = [T_A, T_B, T_C, T_D, T_E, T_F, T_G, T_H]^T$  を求め、次にこのパラメータから焦点距離  $f$ 、原点  $P$ 、方向余弦  $L, M$  などが求まる。

式(A.12)、式(A.13)、式(A.14)より

$$\begin{cases} T_A - X_i^I + \alpha_i T_C + \beta_i T_D - \alpha_i X_i^I T_G - \beta_i X_i^I T_H = 0 \\ T_B - Y_i^I + \alpha_i T_E + \beta_i T_F - \alpha_i Y_i^I T_G - \beta_i Y_i^I T_H = 0 \end{cases} \quad (\text{A.15})$$

( $i = 1, 2, 3, 4$ )

よって、

$$\begin{bmatrix} 1 & 0 & \alpha_1 & \beta_1 & 0 & 0 & -\alpha_1 X_1^I & -\beta_1 X_1^I \\ 0 & 1 & 0 & 0 & \alpha_1 & \beta_1 & -\alpha_1 Y_1^I & -\beta_1 Y_1^I \\ 1 & 0 & \alpha_2 & \beta_2 & 0 & 0 & -\alpha_2 X_2^I & -\beta_2 X_2^I \\ 0 & 1 & 0 & 0 & \alpha_2 & \beta_2 & -\alpha_2 Y_2^I & -\beta_2 Y_2^I \\ 1 & 0 & \alpha_3 & \beta_3 & 0 & 0 & -\alpha_3 X_3^I & -\beta_3 X_3^I \\ 0 & 1 & 0 & 0 & \alpha_3 & \beta_3 & -\alpha_3 Y_3^I & -\beta_3 Y_3^I \\ 1 & 0 & \alpha_4 & \beta_4 & 0 & 0 & -\alpha_4 X_4^I & -\beta_4 X_4^I \\ 0 & 1 & 0 & 0 & \alpha_4 & \beta_4 & -\alpha_4 Y_4^I & -\beta_4 Y_4^I \end{bmatrix} \begin{bmatrix} T_A \\ T_B \\ T_C \\ T_D \\ T_E \\ T_F \\ T_G \\ T_H \end{bmatrix} = \begin{bmatrix} X_1^I \\ Y_1^I \\ X_2^I \\ Y_2^I \\ X_3^I \\ Y_3^I \\ X_4^I \\ Y_4^I \end{bmatrix} \quad (\text{A.16})$$

ここで、頂点1を座標系  $(\alpha, \beta)$  の原点  $(\alpha_1, \beta_1) = (0, 0)$  とすると次のようになる。

$$T_A = X_1^I \quad (\text{A.17})$$

$$T_B = Y_1^I \quad (\text{A.18})$$

$$\begin{bmatrix} \alpha_2 & \beta_2 & 0 & 0 & -\alpha_2 X_2^I & -\beta_2 X_2^I \\ 0 & 0 & \alpha_2 & \beta_2 & -\alpha_2 Y_2^I & -\beta_2 Y_2^I \\ \alpha_3 & \beta_3 & 0 & 0 & -\alpha_3 X_3^I & -\beta_3 X_3^I \\ 0 & 0 & \alpha_3 & \beta_3 & -\alpha_3 Y_3^I & -\beta_3 Y_3^I \\ \alpha_4 & \beta_4 & 0 & 0 & -\alpha_4 X_4^I & -\beta_4 X_4^I \\ 0 & 0 & \alpha_4 & \beta_4 & -\alpha_4 Y_4^I & -\beta_4 Y_4^I \end{bmatrix} \begin{bmatrix} T_C \\ T_D \\ T_E \\ T_F \\ T_G \\ T_H \end{bmatrix} = \begin{bmatrix} X_2^I - X_1^I \\ Y_2^I - Y_1^I \\ X_3^I - X_1^I \\ Y_3^I - Y_1^I \\ X_4^I - X_1^I \\ Y_4^I - Y_1^I \end{bmatrix} \quad (\text{A.19})$$

これを次のようにおく。

$$VT = Z \quad (\text{A.20})$$

ここで、カメラと特徴点4点との相対的な位置関係が一意に決定されるためには、式 (A.20) の  $V$  が次の条件を満たすときである。

$$\det |V| \neq 0 \quad (\text{A.21})$$

この条件のもとで、それら4点の幾何学的な位置関係が既知である場合、カメラのパラメータ、特徴点の画像上での座標（画素データ）、及び特徴点の幾何学的な知識を用いて、対象物に対するカメラ（手先）の位置及び姿勢が一意に定まる。



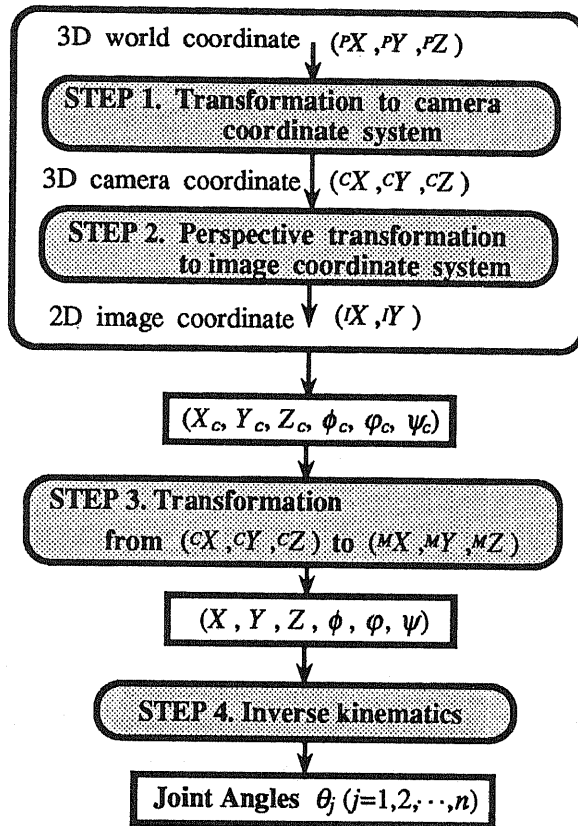


Figure A.3 Transformation of coordinate system

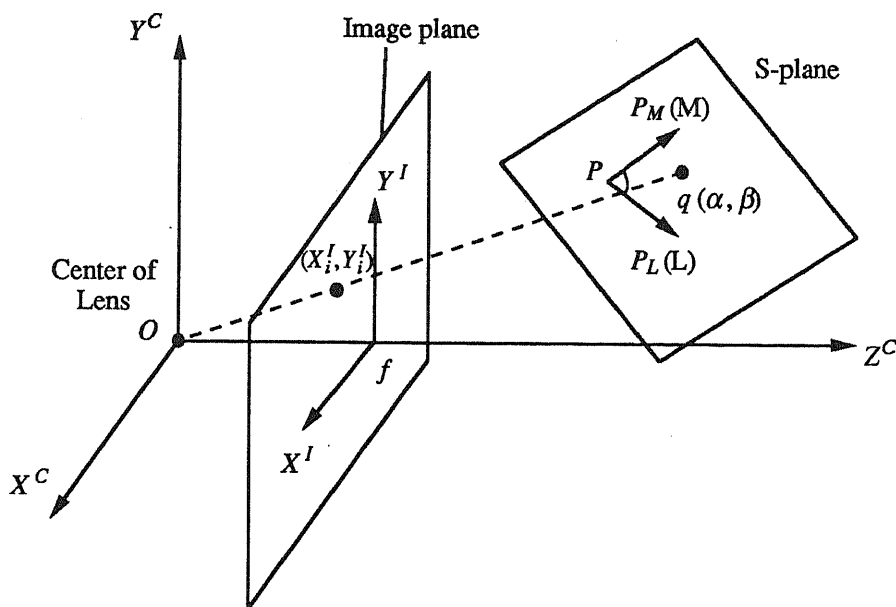


Figure A.4 Coordinate system

## A.3 位置・姿勢制御法

Figure A.5 に示すように、視覚情報を直接利用してマニピュレータの先端を目標の位置・姿勢に移動させる方法を提案する [橋本 90]。

### A.3.1 ビジュアルコントロール

第 A.2.4 節で述べたように対象物の特徴点 4 点が視覚によって認識可能で、それら 4 点の幾何学的な位置関係が既知である場合、カメラのパラメータ、特徴点の画像上での座標（画素データ）、及び特徴点の幾何学的な知識を用いて、対象物に対するカメラ（手先）の位置及び姿勢が一意に定まるということが知られている。このことを利用して、次のビジュアルコントロール法を提案する [Hashimoto89a]。

本研究では、対象物とマニピュレータの先端との相対的な位置関係は未知であるが、対象物の 4 つの特徴点がすべてカメラによって検出可能で、また画像上で認識・識別可能であり、さらに目標位置・姿勢における対象物の画像があらかじめ得られているものと仮定する。Figure A.6 は、ロボットの手先に付けたカメラから得た画像を示している。Figure A.6(a) は、現在の位置及び姿勢における対象物の画像であり、Figure A.6(b) はマニピュレータの先端が、目標とする位置及び姿勢に達したとき得られる画像を示している。

マニピュレータの先端を目標位置、目標姿勢に移動させるためには、初期画像 (Figure A.6(a)) の特徴点 4 点を目標画像 (Figure A.6(b)) の特徴点 4 点に一致させるように各関節への制御入力を生成すればよい。即ち、

$$(A \Rightarrow A^*, B \Rightarrow B^*, C \Rightarrow C^*, D \Rightarrow D^*)$$

となるような関節角  $\Delta\theta$  を求めればよい。

ここで、画像上でのデータ間の差  $\Delta s(\Delta X_i = X_i^* - X_i, \Delta Y_i = Y_i^* - Y_i)$  と関節角空間  $\Delta\theta$  との幾何学的な関係は次のように表される。

$$\Delta s = M(\Delta\theta) \quad (\text{A.22})$$

ただし、 $M(\cdot)$  はマニピュレータの形態によって決まる関数である。したがって、

$$\Delta\theta = M^{-1}(\Delta s) \quad (\text{A.23})$$

画像上でのデータ間の差  $\Delta s$  とマニピュレータの各関節への指令値  $\Delta\theta$  との間の関係は非線形であり、第 A.2.3 節で示した方程式群を解かなければならない。そこで、この非線形な関係をニューラルネットワークを用いて表現する。( Figure A.7 参照)

$$\Delta\theta = NN(\Delta s) \quad (\text{A.24})$$

ただし、 $NN$  はニューラルネットワークを示す。

### A.3.2 ニューラルネットワーク

ここでは、入力データと出力データとの対応付けを学習することができる PDP ネットを用いる [Rumelhart86]。  $m$  入力  $n$  出力の階層型神経回路網は次のように表される。

#### 【階層型ニューラルネット (PDP)】

階層型ニューラルネットワークは、入力ベクトル  $x \in [0, 1]^m$  と出力ベクトル  $y \in [0, 1]^n$  の組が存在したとき  $NN: x \mapsto y$  となる写像を学習によって実現する装置である。

$[0, 1]^m$  を  $(0, 1)$  に写像する任意の連続関数を 3 層のニューラルネットワークで十分多数のニューロンをもつ中間層を用いれば、いくらでも近似できることが知られている。また、階層型ニューラルネットワークは次の補間能力を有することが知られている。

#### 【補間能力】

未学習点に関して今までの学習の結果を反映した出力をだす。

いま、学習点を  $(x_0, y_0)$  とすると、 $\forall \varepsilon > 0, \exists \delta \forall x \in \{x \mid \|x - x_0\| < \delta\}$  に対して  $y = NN(x), \|y - y_0\| < \varepsilon$  となる。すなわち、学習した領域内で一様連続性を保証する。

ニューラルネットワークは、Figure A.8 に示すように入力層、隠れ層、出力層の三層構造からなり、入力層には、初期画像と目標画像における特徴点4点の座標偏差 $(\Delta X_i, \Delta Y_i)$ を入力し、出力層には出力すべきロボットマニピュレータの各関節の角度情報 $(\Delta \theta_j)$ を教師信号として学習を繰り返す。ここでは、下記のエネルギー関数を0になるように重みを変更して学習する。

$$E = \sum_k (\Delta \theta_{k.des} - \Delta \theta_{k.act})(\Delta \theta_{k.des} - \Delta \theta_{k.act})^T \quad (\text{A.25})$$

ここで学習パターンを次のように表す。

$$PT^k = (\Delta s^k, \Delta \theta^k) \quad (\text{A.26})$$

入力・出力パターンの関係を学習したニューラルネットワークは、マニピュレータの先端と対象物との間の位置・姿勢の関係をモデル化したことになる。

### A.3.3 学習・実行プロセス

Figure A.9 に学習プロセスを示す。学習プロセスでは、マニピュレータの先端を目標位置・姿勢から実際にランダムに動かし、初期画像を得る。この得られた初期画像と目標画像の2枚の画像データから特徴点4点の画像上での座標の偏差を算出し、ニューラルネットワークの入力層に入力する。このとき、各関節に付けた内界センサから得られる各関節の角度情報を教師信号として与える。そして、教師信号とニューラルネットワークの出力値との差が0に近づくまでバックプロパゲーションによる学習を繰り返す。

Figure A.10 に実行プロセスを示す。実行プロセスでは、学習の終わったニューラルネットワークを用いて、現画像（初期画像）と目標画像上の特徴点（4点）が一致するのに必要な各関節のコントロール情報を生成する。そして、実際にその情報に基づいてマニピュレータを動かす。

### A.3.4 学習の高速化

バックプロパゲーションによる学習において学習時間が問題となっており、さまざまな高速化アルゴリズムが提案されている。ここでは、以下の否定入力の提示による高速化手法を用いる [橋本 88]。学習パターンによって大きく異なる収束性が、全体の学習の収束性に大きく影響を及ぼすと考えられる。入力パターン  $U_i$  は、 $U_i^2$  という形でエネルギーの減少速度に関係している。入力パターン  $U_i$  を規格化して考えると  $U_i$  の値が 0 から 1 まで変化すると  $U_i^2$  も 0 から 1 まで大きく変化してしまい、入力されるパターンによってかなりのむらが生じることになる。これを平滑化するために  $U_i$  と同時に  $1 - U_i$  を入力すると

$$U_i^2 + (1 - U_i)^2 = 2(U_i - 0.5)^2 + 0.5 \quad (\text{A.27})$$

の形でエネルギーの減少速度に寄与することになり、 $U_i$  の値が 0 から 1 まで変化してもその影響は 0.5 から 1 になり、パターンの違いによるエネルギーの減少速度の偏りを是正することができる。入力  $1 - U$  は、入力  $U$  の否定とみなすことができるので否定入力の提示とよばれている。この手法はエネルギー関数の形状を変えているという見方もできる。すなわち、否定入力を提示することによりエネルギー関数が収束し易い形状になっていると考えられる。

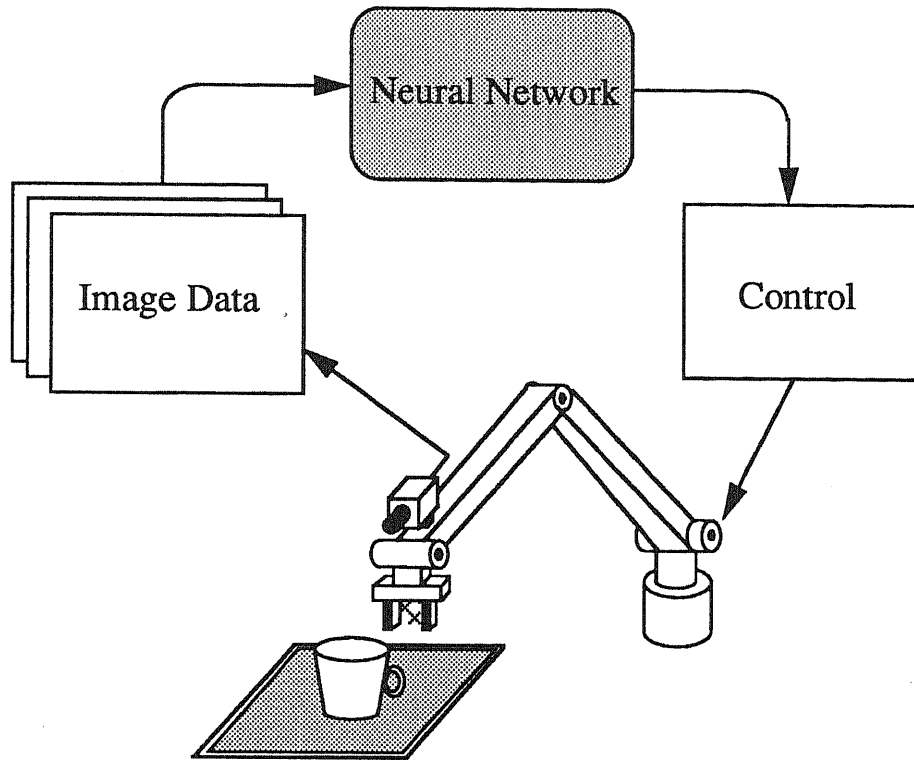


Figure A.5 Visual control system

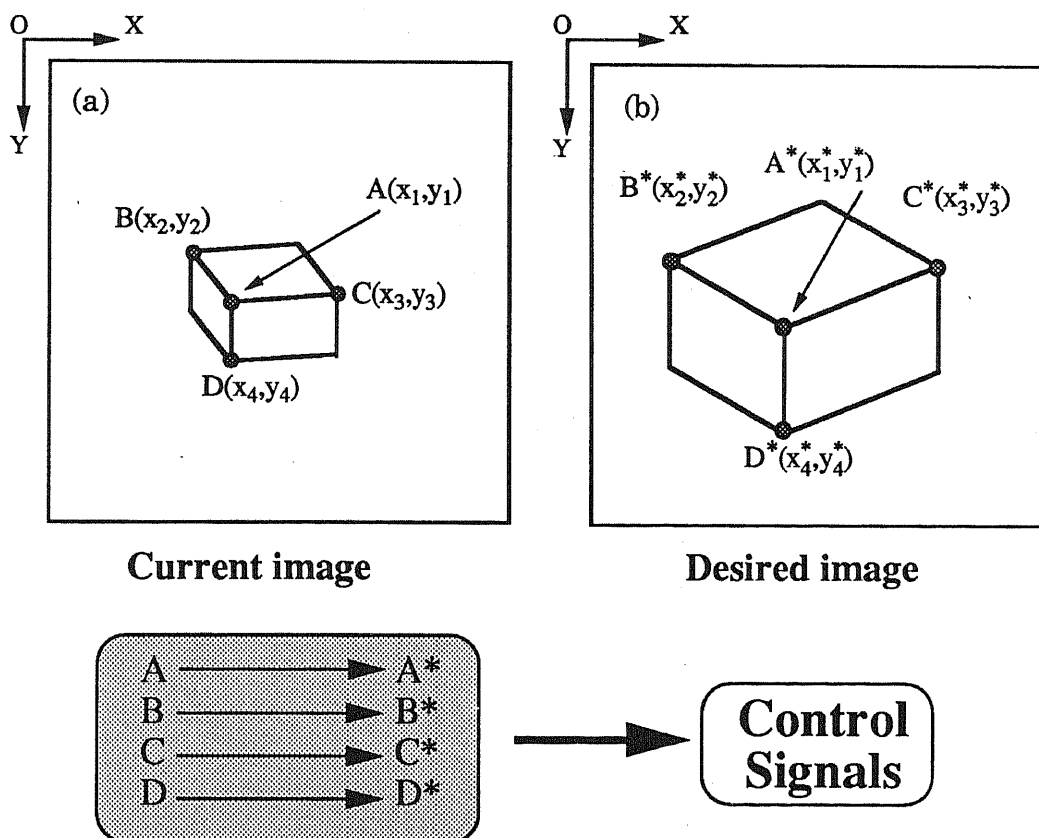


Figure A.6 Camera image

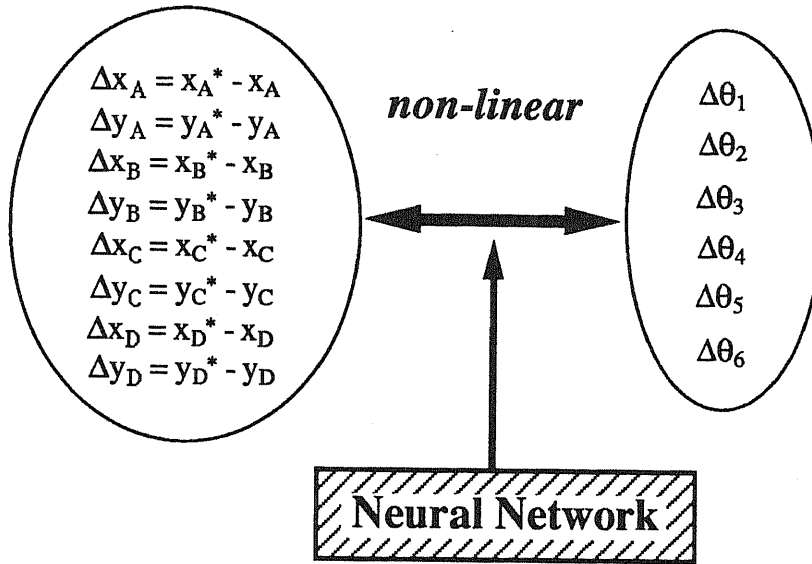


Figure A.7 Image data and control signals

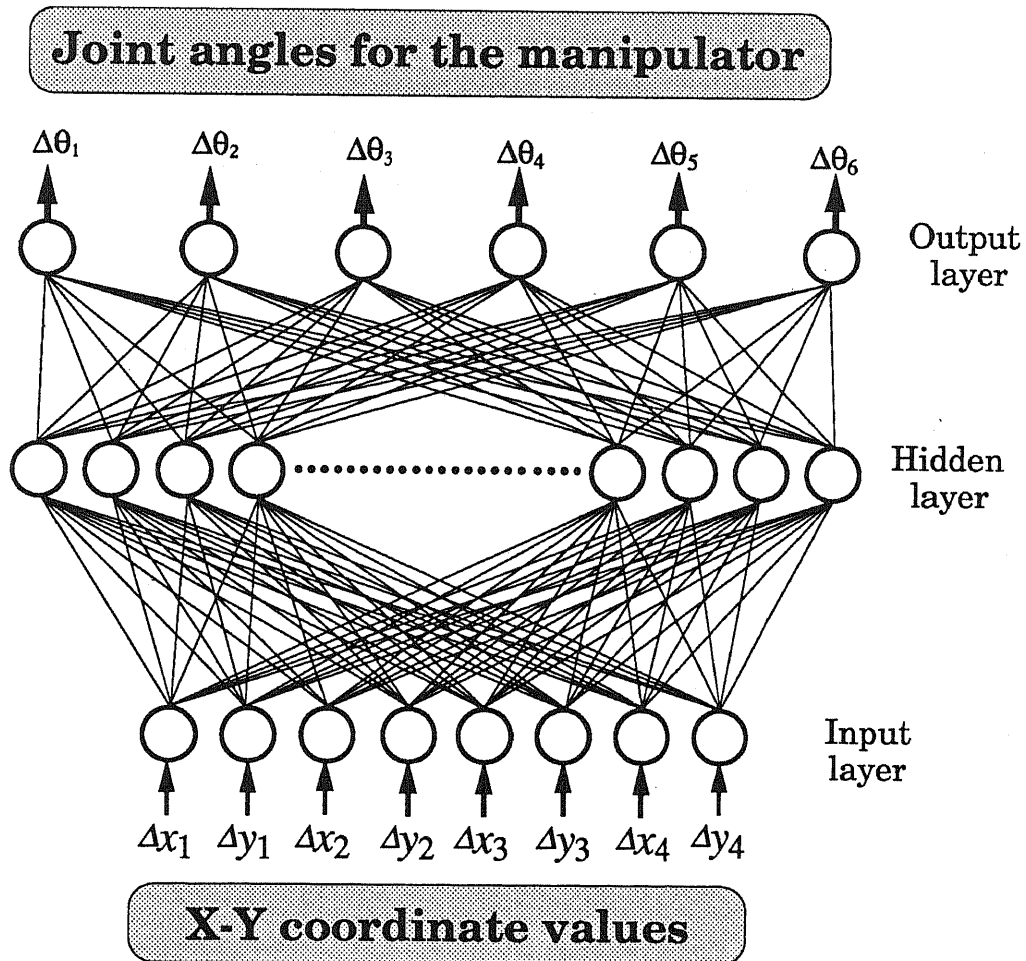


Figure A.8 PDP network

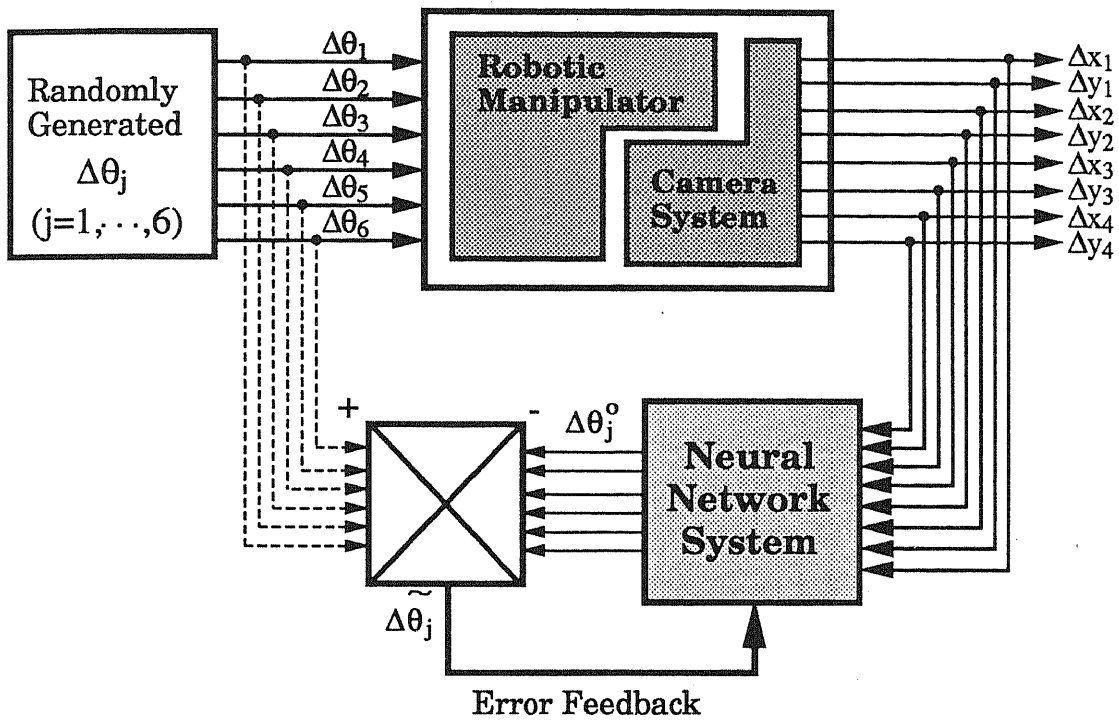


Figure A.9 Block diagram of learning process

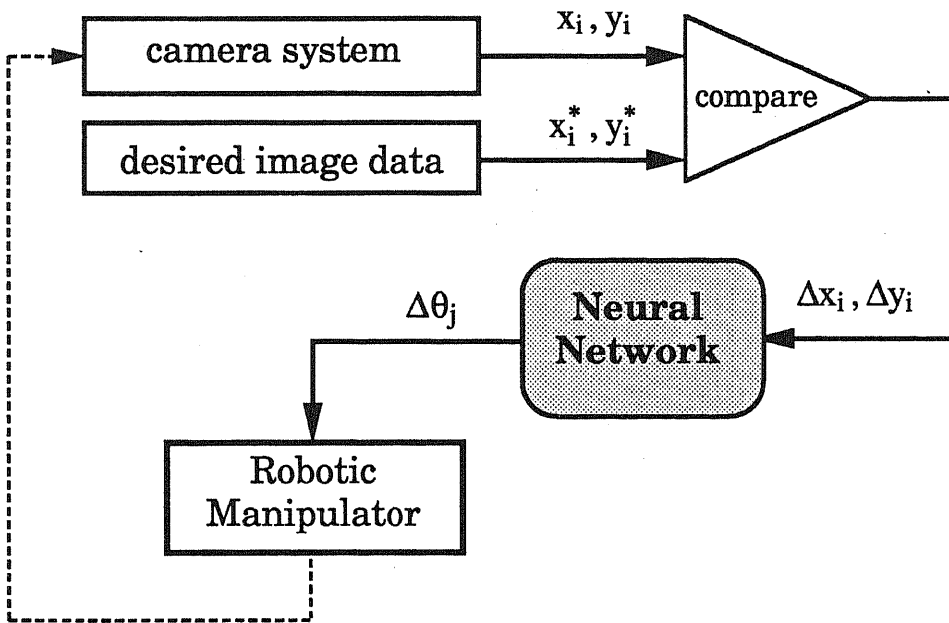


Figure A.10 Block diagram of execution process



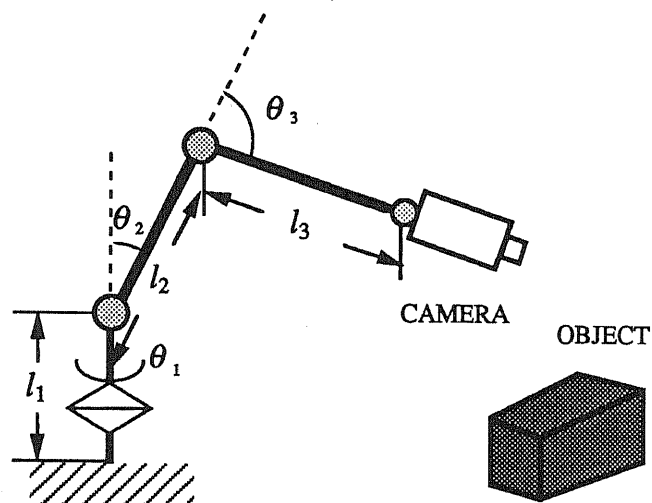
### A.3.5 シミュレーションによる考察

#### シミュレーション1 (位置制御)

まず、本手法を Figure A.11 に示す 3 自由度マニピュレータの位置制御に対して適用し、シミュレーションを行った。シミュレーションでは、Figure A.12 に示すように、入力層 8 ユニット、隠れ層 12 ユニット、出力層 3 ユニットの 3 層構造から成る PDP ネットを用いた。なお、シミュレータとして NeuralWorks を用いており、ニューラルネットの初期値等はランダムに設定できる。学習則は、通常のバックプロパゲーションを用いた。学習プロセスでは Table A.1 に示す 30 の学習パターンを繰り返し教え、約 500 回の学習回数で収束したが、精度を上げるためにさらに約 10,000 回の学習を行った。Figure A.13(1a)(1b)(1c) は学習させたパターンに対するもので、Figure A.13(2a)(2b)(2c) は学習していないパターンに対するものである。マニピュレータの先端は、視覚情報のみから目標の位置にかなり接近し、ほぼ一致している。また、教えていない目標パターンに対しても同様の結果が得られた。

#### シミュレーション2 (姿勢制御)

本手法を 6 自由度マニピュレータの姿勢制御に対して適用し、シミュレーションを行った。ただし、ここでは第 1 関節、第 2 関節、第 3 関節を固定している。シミュレーションに用いた PDP ネットの構成は、8 入力、6 出力である。中間層の個数は試行錯誤を繰り返し、最終的に 24 個用いた。学習プロセスでは 35 の学習パターンを繰り返し教え、約 1,000 回の学習回数で収束したが、精度を上げるためにさらに約 10,000 万回の学習を行った。Figure A.14(1a)(1b) は学習させたパターンに対するもので、Figure A.14(2a)(2b)、Figure A.14(3a)(3b) は学習していないパターンに対するものである。いずれの場合にもマニピュレータの先端は、目標の姿勢にかなり接近し、ほぼ一致している。



**Model**

$$x = \{ l_2 \sin \theta_2 + l_3 \sin (\theta_2 + \theta_3) \} \cos \theta_1$$

$$y = \{ l_2 \sin \theta_2 + l_3 \sin (\theta_2 + \theta_3) \} \sin \theta_1$$

$$z = l_1 + l_2 \cos \theta_2 + l_3 \cos (\theta_2 + \theta_3)$$

Figure A.11 Robotic manipulator with three degrees of freedom

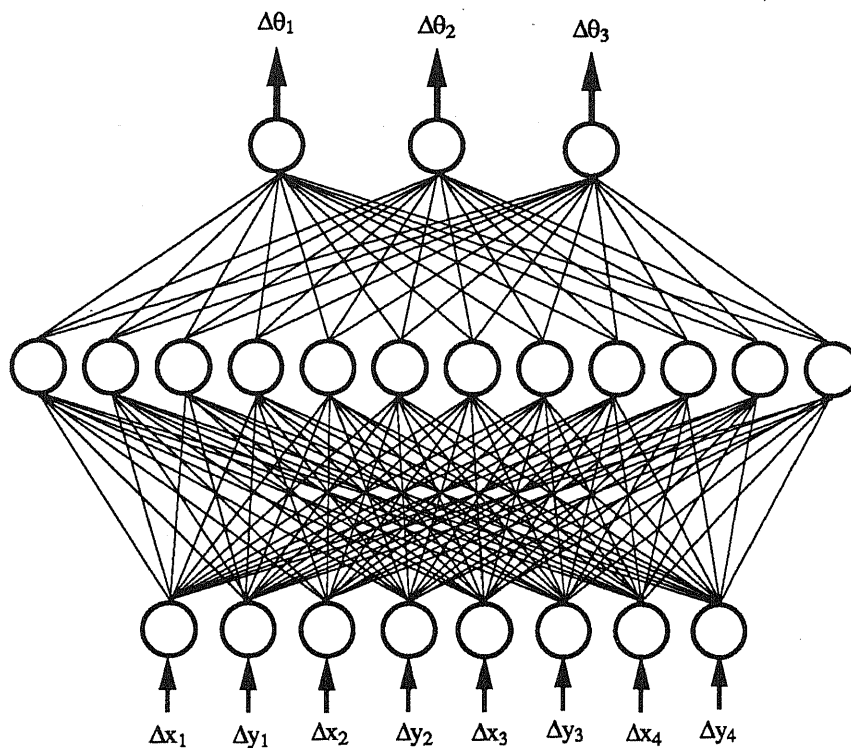


Figure A.12 Neural network used for simulation(1)

Table A.1 Training patterns for simulation (1)

$\Delta\theta_1$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$\Delta\theta_2$	5.0	10.0	15.0	20.0	25.0	30.0	35.0	40.0	45.0	48.0
$\Delta\theta_3$	-5.0	-10.0	-15.0	-20.0	-24.0	-29.0	-34.0	-39.0	-44.0	-47.0

$\Delta\theta_1$	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0
$\Delta\theta_2$	0.0	5.0	10.0	15.0	20.0	25.0	30.0	35.0	40.0	45.0
$\Delta\theta_3$	0.0	-5.0	-10.0	-14.0	-19.0	-24.0	-29.0	-34.0	-39.0	-44.0

$\Delta\theta_1$	-2.0	-4.0	-4.0	-4.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0
$\Delta\theta_2$	50.0	0.0	5.0	10.0	20.0	25.0	30.0	35.0	40.0	45.0
$\Delta\theta_3$	-50.0	0.0	-5.0	-10.0	-20.0	-25.0	-30.0	-35.0	-40.0	-45.0

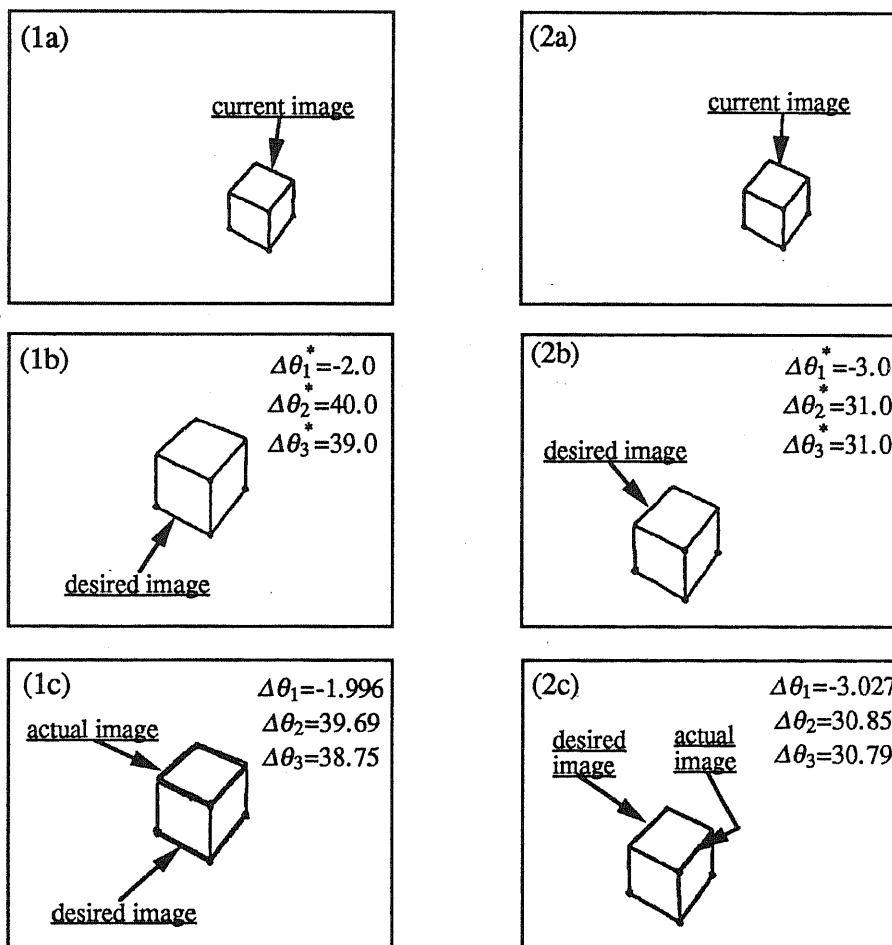


Figure A.13 Simulation results

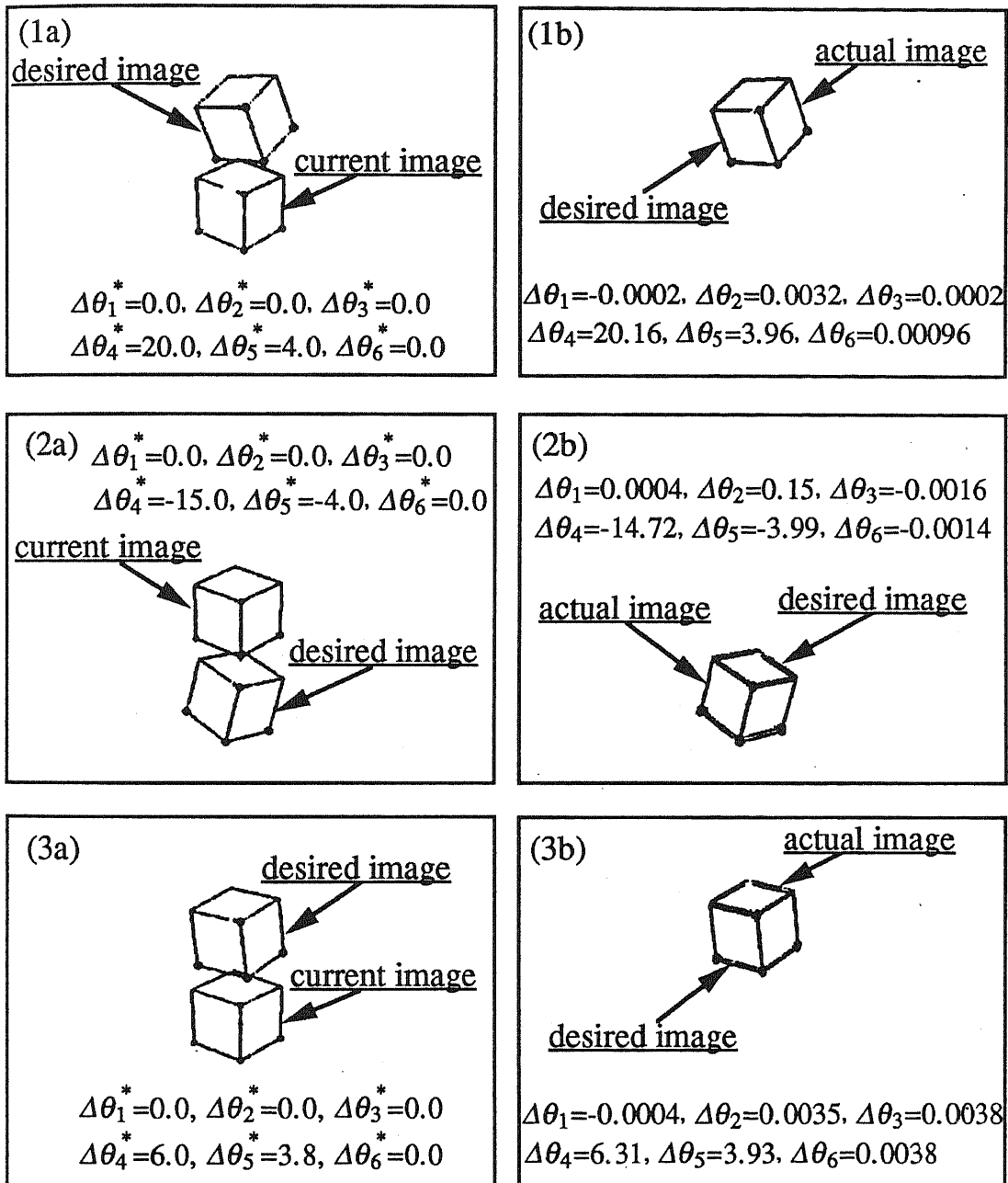


Figure A.14 Simulation results

## シミュレーション3 (位置・姿勢制御)

本手法を6自由度マニピュレータの位置・姿勢制御に対して適用し、シミュレーションを行った。Figure A.15 にシミュレーションに用いた6自由度マニピュレータのモデルを示す。またシミュレーションでは、入力層8ユニット、隠れ層48ユニット、出力層6ユニットの3層構造から成るPDPネットを用いた。

## 6自由度マニピュレータの幾何学モデル

この同次変換行列  $T_E^R$  は次のように表される。

$$T_E^R = \begin{bmatrix} R_{11} & R_{12} & R_{13} & p_x \\ R_{21} & R_{22} & R_{23} & p_y \\ R_{31} & R_{32} & R_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.28})$$

ただし、ここで

$$S_{ijk} = \sin(\theta_i + \theta_j + \theta_k) \quad C_{ijk} = \cos(\theta_i + \theta_j + \theta_k) \quad (\text{A.29})$$

とすると

$$\begin{aligned} R_{11} &= C_1[C_{23}(C_4C_5C_6 - S_4S_6) - S_{23}S_5C_6] + S_1(S_4C_5C_6 + C_4S_6) \\ R_{12} &= C_1[-C_{23}(C_4C_5S_6 + S_4C_6) + S_{23}S_5S_6] - S_1(S_4C_5S_6 - C_4C_6) \\ R_{13} &= -C_1(C_{23}C_4S_5 + S_{23}C_5) - S_1S_4S_5 \\ R_{21} &= S_1[C_{23}(C_4C_5C_6 - S_4S_6) - S_{23}S_5C_6] - C_1(S_4C_5C_6 + C_4S_6) \\ R_{22} &= S_1[-C_{23}(C_4C_5S_6 + S_4C_6) + S_{23}S_5S_6] + C_1(S_4C_5S_6 - C_4C_6) \\ R_{23} &= -S_1(C_{23}C_4S_5 + S_{23}C_5) + C_1S_4S_5 \\ R_{31} &= -S_{23}(C_4C_5C_6 - S_4S_6) - C_{23}S_5C_6 \\ R_{32} &= S_{23}(C_4C_5S_6 + S_4C_6) + C_{23}S_5S_6 \\ R_{33} &= S_{23}C_4S_5 - C_{23}C_5 \\ p_x &= C_1(l_cC_2 + l_eC_{23} - l_fS_{23}) - (l_b - l_d)S_1 \\ p_y &= S_1(l_cC_2 + l_eC_{23} - l_fS_{23}) + (l_b - l_d)C_1 \\ p_z &= -l_cS_2 - l_eS_{23} - l_fC_{23} \end{aligned}$$

式(A.28)よりマニピュレータの手先位置ベクトル  $r$  は次のように求まる。

$$r = [r_1, r_2, r_3, r_4, r_5, r_6]^T \quad (\text{A.30})$$

ただし、

$$r_1 = p_x + R_{13}l_g$$

$$r_2 = p_y + R_{23}l_g$$

$$r_3 = p_z + l_a + R_{33}l_g$$

$$r_4 = \text{atan2}(R_{23}, R_{13})$$

$$r_5 = \text{atan2}(\sqrt{R_{13}^2 + R_{23}^2}, R_{33})$$

$$r_6 = \begin{cases} \text{atan2}(R_{32}, -R_{31}) & (R_{13}^2 + R_{23}^2 \neq 0 \text{ のとき}) \\ \text{atan2}(R_{21}, R_{22}) - R_{33}r_4 & (R_{13}^2 + R_{23}^2 = 0 \text{ のとき}) \end{cases}$$

### 【学習プロセス】

カメラ画像の左上の座標値を (0, 0) としたとき、目標画像での特徴点 4 点の座標値を次に示す。

$$A^* = (192, 207), \quad B^* = (91, 120)$$

$$C^* = (293, 120), \quad D^* = (192, 274)$$

カメラ視野の制限により、特徴点 4 点すべてがカメラの視野内にはいるように、マニピュレータの手先（カメラ）の動作範囲を Figure A.16 に示す範囲内に設定してシミュレーションを行う。学習データとしては、各関節角度を 1 度刻みにランダムに発生させ、60 パターンの教師信号を得る。そして、それぞれのパターンでの関節角度をマニピュレータに入力し、その時得られる画像データ（初期画像）と目標画像との偏差を計算してニューラルネットワークへの入力とする。ニューラルネットワークへの教師信号は各関節角度である。Table A.2 に 60 パターンのうちのいくつかのパターン例を示す。これら 60 パターンのデータを繰り返し教え、約 3 万回の学習を行った。

### 【実行プロセス】

実行プロセスでは、学習プロセスにおいて 60 パターンのデータを学習したニューラルネットワークを用いて、実際にニューラルネットワークからの出力値を制御入力と

してマニピュレータを動かす。Figure A.17 は、学習させたパターンに対するシミュレーション結果を示したものであり、目標画像と一致することがわかる。そのときの特徴点4点についての画素エラーを Table A.3 に示す。Table A.3 の上段 (Initial Error) は、目標画像と初期画像における特徴点4点の画素の偏差をとったものであり、下段 (Final Error) は、目標画像とニューラルネットからの出力をマニピュレータに入力して動かしたときに得られた画像の特徴点4点の画素エラーを表したものである。このデータより最終エラーは  $-4 \sim +4$  画素程度に抑えられた。この結果より、学習したパターンについては、マニピュレータの先端を目標とする位置・姿勢にほぼ移動することができる。

### 【未学習パターンへの適用】

ここでは、ニューラルネットワークの補間能力を調べるために、先に学習した60パターン以外のデータについて適用した場合についてシミュレーションを行った。ここでの未学習パターンとは学習していない任意の初期位置のことを指す。データとしては、学習パターンと同様に Figure A.16 に示した範囲内でランダムに発生させた。そのデータを実行プロセスに従って実行する。そのときのシミュレーション結果を Figure A.18 に示す。また、そのときの画素エラーを Table A.4 に示す。学習したパターンに比べると、目標画像に完全に一致はしていないが、大まかに位置・姿勢の制御を行うことができる。しかしながら、Figure A.19(1a)(1b) に示すように目標画像から多少ずれてしまうものもある。この原因としては、学習パターンが少ないために画像データと関節角との非線形関係が十分に表されていないことが考えられ、このような問題に対しては、学習パターンの数を多くすることが考えられる。また、目標の近傍を詳細に教えた別のネットワークを用いて、さらにずれ (エラー) をフィードバックするという方法を用いることによって精度を向上させることができると思われる。これに関しては第 A.4 節で述べる。以上のことより、対象物の特徴点4点すべてが検出・認識・識別可能であるという条件の下で、マニピュレータの先端を、視覚情報とニューラルネットワークを用いて目標の位置・姿勢にかなり接近させ、ほぼ一致させることができる。

ここでは、対象物の特徴点4点がすべてカメラによって検出可能で、画像上でその4点が認識・識別できるという条件のもとに、作業空間を限定しているが、ニューラルネットワークを用いて、広大な作業空間を写像しようとする場合には、入出力において0から1 (連続値) に規格化することによる限界、カメラ視野の問題、さらにはオクルージョンなどの問題がある。このような場合には、作業空間を分割して複数のニューラルネットワークを用いるなどの工夫が必要になると考えられる。

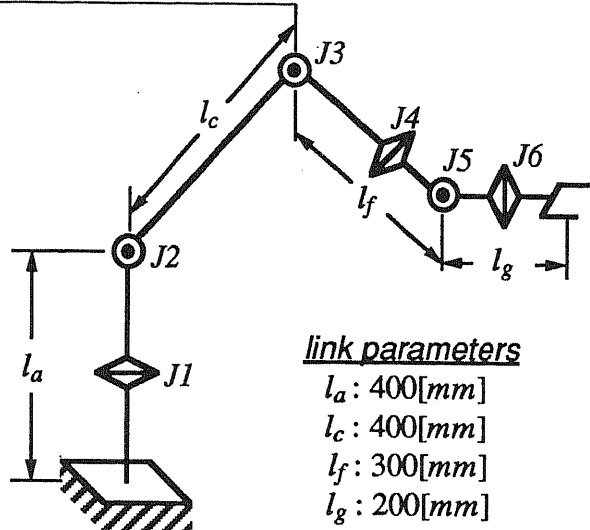


Figure A.15 Simulation model

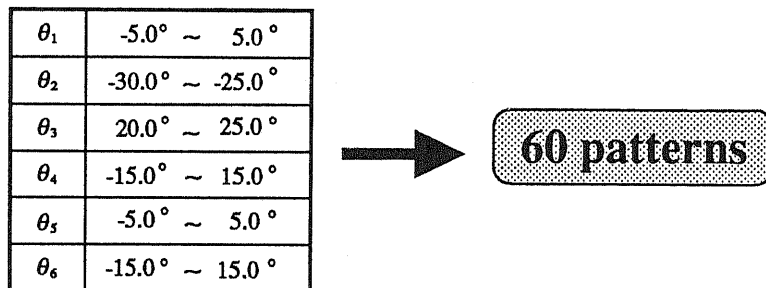


Figure A.16 Restrictions to training patterns

Table A.2 Training patterns

Trainig No.	Input Patterns (pixel)								Output Patterns [°]					
	$\Delta x_1$	$\Delta y_1$	$\Delta x_2$	$\Delta y_2$	$\Delta x_3$	$\Delta y_3$	$\Delta x_4$	$\Delta y_4$	$\Delta \theta_1$	$\Delta \theta_2$	$\Delta \theta_3$	$\Delta \theta_4$	$\Delta \theta_5$	$\Delta \theta_6$
1	-115	6	-108	26	-137	-81	-51	53	-5.0	-29.0	23.0	-13.0	-3.0	-5.0
2	18	-78	42	-55	15	14	108	-38	2.0	-28.0	22.0	-15.0	4.0	-9.0
3	28	-20	29	-2	-14	39	71	40	1.0	-30.0	25.0	-13.0	-2.0	14.0
4	-84	-146	-91	-135	-129	-98	-53	-94	-4.0	-26.0	24.0	-10.0	2.0	9.0
5	-123	-2	-116	16	-141	72	-59	40	-5.0	-26.0	21.0	-8.0	-3.0	-11.0
6	-68	-89	-57	-70	-87	-9	3	-45	-3.0	-30.0	22.0	-7.0	5.0	-13.0
7	-2	-100	-1	-87	-40	-47	39	-49	0.0	-26.0	23.0	-8.0	1.0	13.0
8	24	2	10	23	-48	46	40	80	0.0	-28.0	22.0	4.0	-1.0	13.0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
60	-84	-140	-99	-127	-143	-100	-69	-80	-4.0	-25.0	22.0	1.0	4.0	5.0



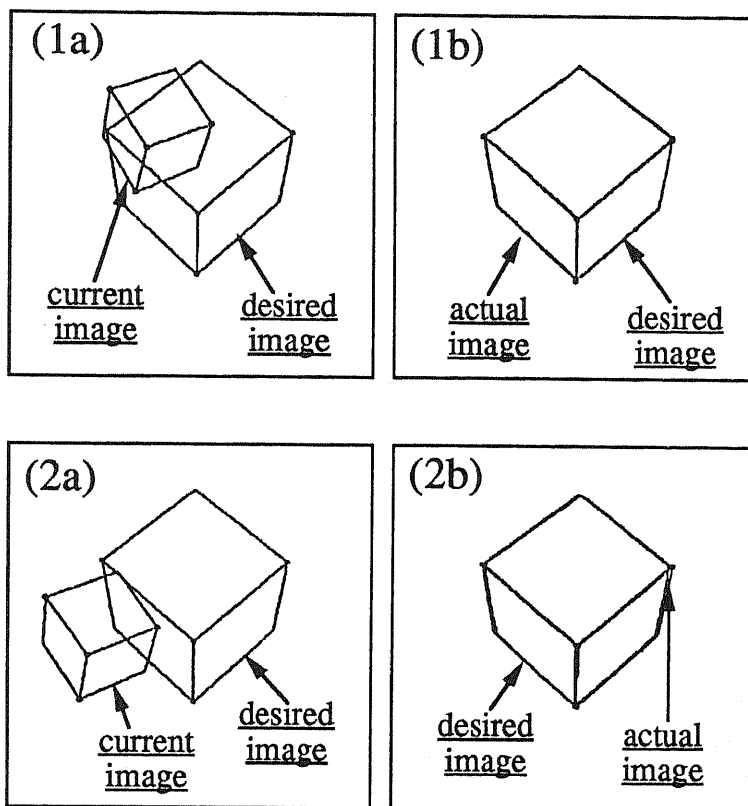


Figure A.17 Simulation results (1) [learned patterns]

Table A.3 Pixel error (learned patterns )

Error	$\Delta x_1$	$\Delta y_1$	$\Delta x_2$	$\Delta y_2$	$\Delta x_3$	$\Delta y_3$	$\Delta x_4$	$\Delta y_4$
Initial	-57	-70	3	-45	-87	-9	-68	-89
Final	-1	2	-1	2	0	3	-1	1
Initial	-116	16	-59	40	-141	72	-123	-2
Final	3	0	3	0	4	0	2	1
Initial	8	51	50	86	-31	90	5	33
Final	-2	1	-4	0	-2	0	-2	1
Initial	-1	-87	39	-49	-40	-47	-2	-100
Final	-1	2	-2	1	0	2	-1	2

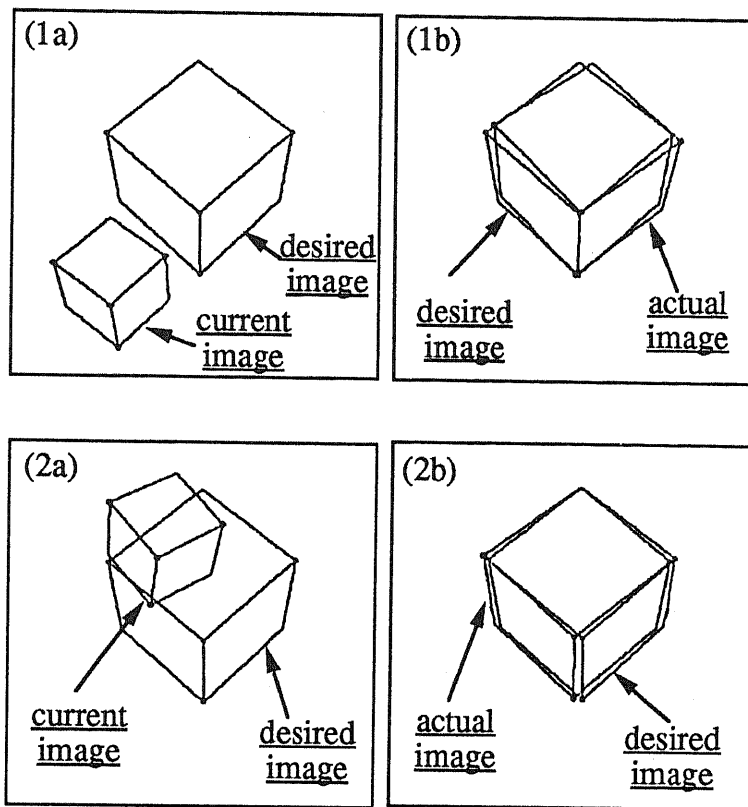


Figure A.18 Simulation results (2) [not learned pattern]

Table A.4 Pixel error (not learned pattern)

Error	$\Delta x_1$	$\Delta y_1$	$\Delta x_2$	$\Delta y_2$	$\Delta x_3$	$\Delta y_3$	$\Delta x_4$	$\Delta y_4$
Initial	-98	101	-56	141	-139	134	-90	80
Final	2	-2	9	-9	9	7	-3	-1
Initial	-49	-91	2	-64	-79	-39	-56	-106
Final	-9	-1	-5	-4	-6	3	-10	-2
Initial	13	-33	48	13	-33	-1	18	-49
Final	-1	-14	-3	-11	-2	-14	0	-11
Initial	52	-15	91	21	13	23	48	-32
Final	-9	5	-12	7	-11	1	-6	3

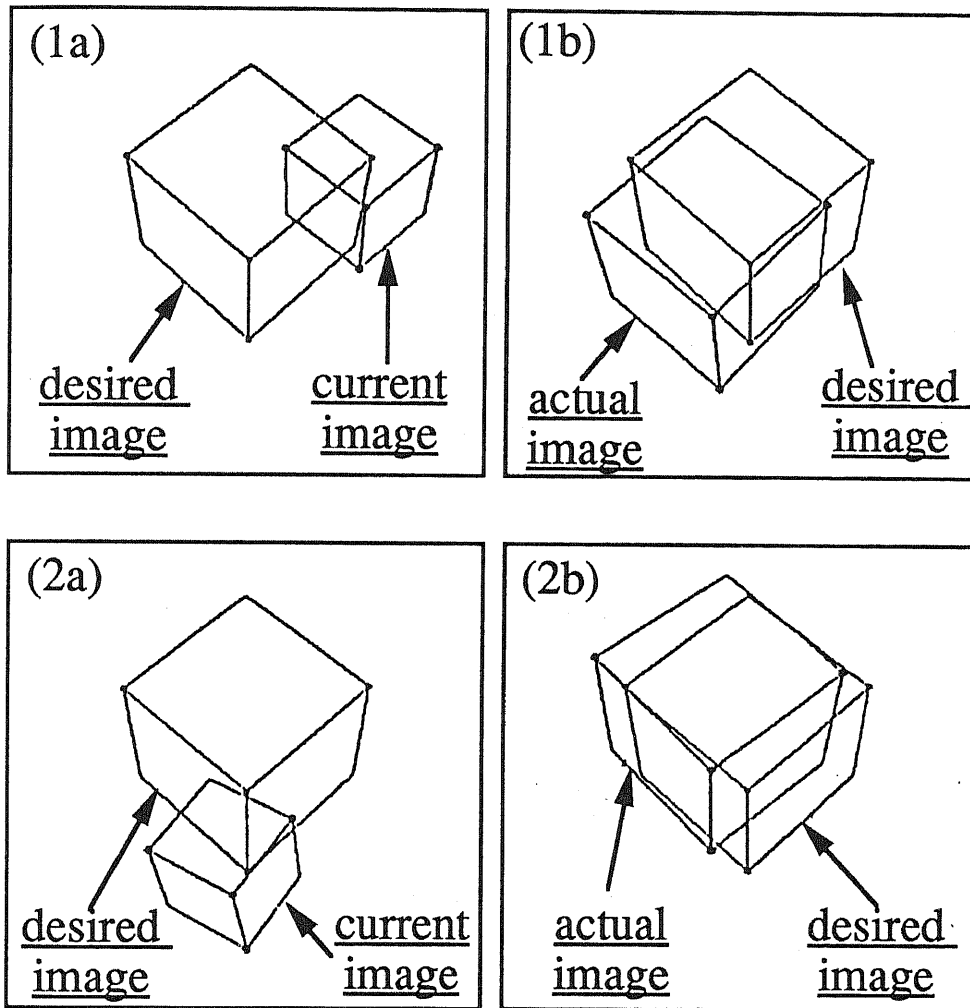


Figure A.19 Simulation results (3) [not learned pattern]

### A.3.6 特徴点4点の識別法

シミュレーションでは、対象物の特徴点4点すべてが検出・認識・識別可能であるという条件の下で、マニピュレータの制御を行っていたが、ここで、従来の画像処理により4点の検出は可能であるが、4点の対応を決める識別方法について述べる。本来では4点の組合せであるので、24通りの可能性がある。ここでは以下の方法を提案する。

#### (1) 問題設定

特徴点は検出可能（カメラ視野内）で4点の候補となる点の画像上の座標は、画像処理（微分画像および探索）によって求められるものとする。また、物体の形状については既知である。このとき、画面上の像から各点がどれに対応しているかを決定する。

#### (2) 識別方法

4点の中で点Aに関しては3面頂点の形（フォースまたはアーク）及び他の点との位置関係から容易に識別可能である。したがって、残りの3通りについて考察する。

Figure A.20(a) は直方体を視点  $P_v$  からみた像をスクリーン平面  $S_p$  上に写したところを表している。また、Figure A.20(b) に示すように直方体の辺は3組の平行な直線の組からなるが、画像上では各直線の延長線上である一点で交わる。Figure A.20(c) のように座標系をとる。さらに、直方体の辺の1つである辺  $AE$  を視点  $P_v$  からみた図を Figure A.20(d) に示す。Figure A.20(b) において直線  $AE$  と直線  $DB$  の延長線が交わった点  $Y_\infty$  は Figure A.20(d) のように視点  $P_v$  からは、直線  $AE$  と平行な方向に見える。したがって、視点  $P_v$  から各点のみえる方向のなす角度として Figure A.20(d) のように  $\rho_{1Y}$ ,  $\rho_{2Y}$  とおけば

$$AQ = AE \cdot \sin \rho_{1Y} = e \cdot \sin \rho_{1Y} \quad (\text{A.31})$$

$$= AP_v \cdot \sin \rho_{2Y} = p \cdot \sin \rho_{2Y} \quad (\text{A.32})$$

となるから

$$p = \frac{e \cdot \sin \rho_{1Y}}{\sin \rho_{2Y}} \quad (\text{A.33})$$

となる。これを辺  $AD$ 、 $AF$  についても同様におこなうことによって次のような式を得る。

$$p = \frac{e \cdot \sin \rho_{1Y}}{\sin \rho_{2Y}} = \frac{d \cdot \sin \rho_{1X}}{\sin \rho_{2X}} = \frac{f \cdot \sin \rho_{1Z}}{\sin \rho_{2Z}} \quad (\text{A.34})$$

このことから、3つの辺  $AE$ 、 $AD$ 、 $AF$  が画面上のどの辺に対応するかは、点  $A$  を端点にもつ画面上の辺は3つしかないのであるから、これらの組合せについて絞ることができる。そこで、可能性のある3つの組合せについて、式(A.34)に代入して成り立つかどうかを調べればよい。これによって残りの3点(点  $D$ 、点  $E$ 、点  $F$ )が決定される。

さらに、3点(点  $D$ 、点  $E$ 、点  $F$ )がわかれば、式(A.34)から  $p$  の大きさも計算できるので、視点  $P_v$  の座標は次のように表される。

$$(-p \cos(\rho_{1X} + \rho_{2X}), -p \cos(\rho_{1Y} + \rho_{2Y}), -p \cos(\rho_{1Z} + \rho_{2Z})) \quad (\text{A.35})$$

このようにして画面上の像から直方体と視点との位置関係を一意に知ることができる。

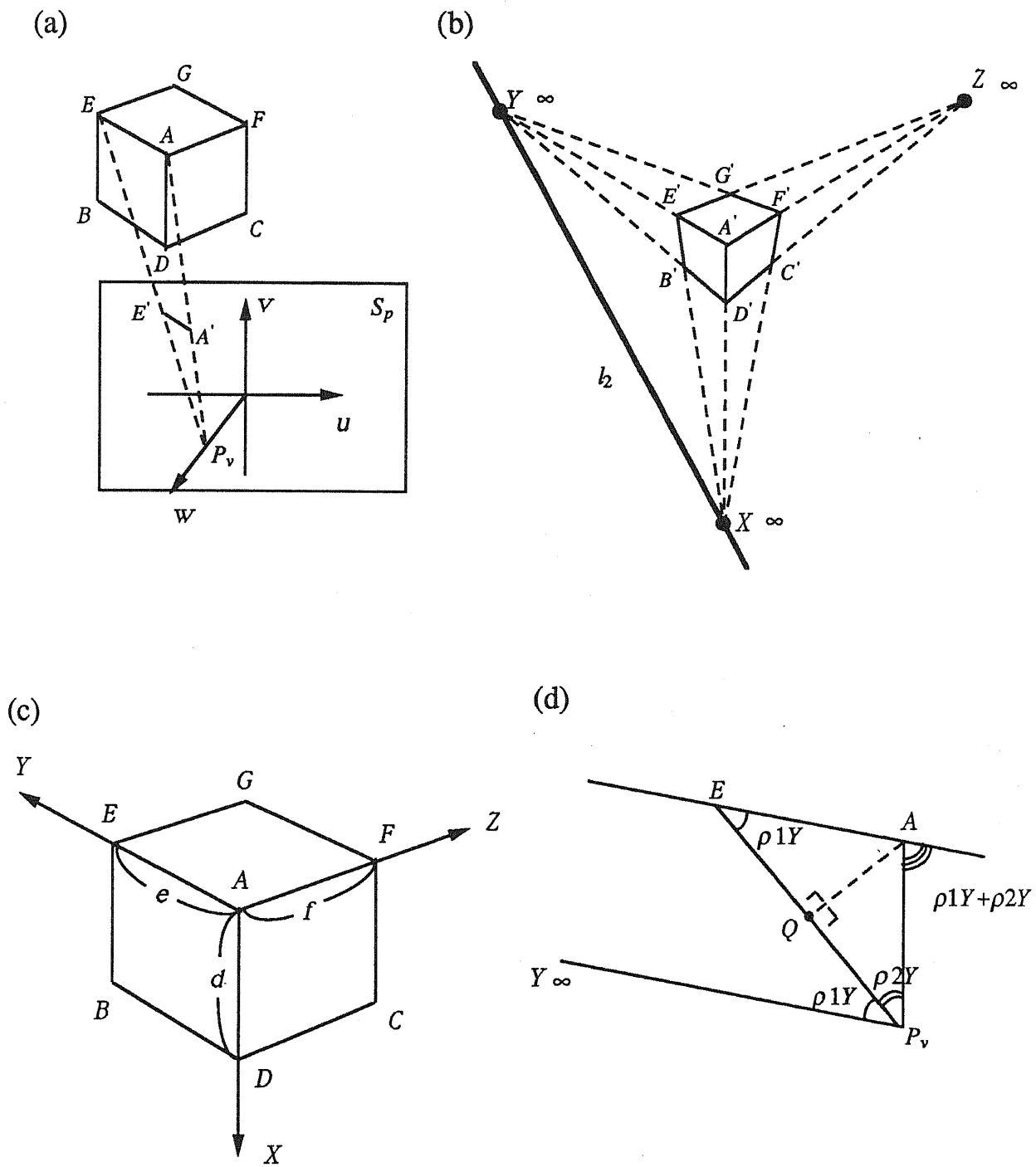


Figure A.20 Relation between image and object

## A.4 ニューラルネットワークシステム

ここでは、第 A.3.5 節のシミュレーション結果及び考察をふまえて、Figure A.21 に示すようにニューラルネットワークシステムとして2種類のネットをルールに基づいて切り換えて、マニピュレータの先端の制御を行う [Hashimoto89c]。

### A.4.1 学習プロセス

Figure A.22 に学習プロセスを示す。ここでは、効率的な学習を行うために、また、広大な作業空間を写像する場合、入出力において0から1（連続値）に規格化することにより生じる限界に対処するために、2種類のネットワークを用いる。1つは大域的な範囲を学習するグローバルニューラルネットワークと、もう1つは目標の近傍の局所的な範囲を学習するローカルニューラルネットワークである。

グローバルニューラルネットワークの構成図を Figure A.23 に示す。グローバルニューラルネットの入力には、特徴点4点の画像上での座標の偏差だけでなく、そのときの関節角度も入力する。マニピュレータの幾何学的な関係がその状態角度の影響を受けるため、ニューラルネットにあらかじめ情報として与えるためである。この状態角度情報は、ある基準からエンコーダ等の内界センサから容易に得られる。

$$\Delta\theta = NN_G(\Delta s, \theta) \quad (\text{A.36})$$

一方、ローカルニューラルネットワークの構成図を Figure A.24 に示す。ローカルネットは目標位置・姿勢の近傍を学習するため、特徴点4点の画像上での座標の偏差だけを入力する。

$$\Delta\theta = NN_L(\Delta s) \quad (\text{A.37})$$

## A.4.2 実行プロセス

Figure A.25 に実行プロセスを示す。実行プロセスでは、学習の終わったニューラルネットワークを用いて、現画像（初期画像）と目標画像上の特徴点（4点）が一致するのに必要な各関節のコントロール情報を生成する。まず、グローバルネットを用いて目標位置・姿勢に近づき、近傍において誤差が生じた場合にローカルネットに切り換えて誤差を吸収するように二段階プロセスで実行する。

グローバルニューラルネットワークとローカルニューラルネットワークとの切り換えルールを以下に示す。（Figure A.26 参照）

[切り換えルール]

IF  $[\forall m, n (m \neq n) \quad \alpha_{mn} \leq \alpha_{th}]$   
 THEN ローカルニューラルネットワーク  
 ELSE グローバルニューラルネットワーク

ただし、

$$\alpha_{mn} = \left| \frac{L_{mn}^* - L_{mn}}{L_{mn}^*} \right|, \quad m, n \in \{A, B, C, D\}$$

- $L_{mn}$  : 点  $m$  と点  $n$  との間の現画像上における距離
- $L_{mn}^*$  : 点  $m$  と点  $n$  との間の目標画像上における距離
- $\alpha_{th}$  : しきい値



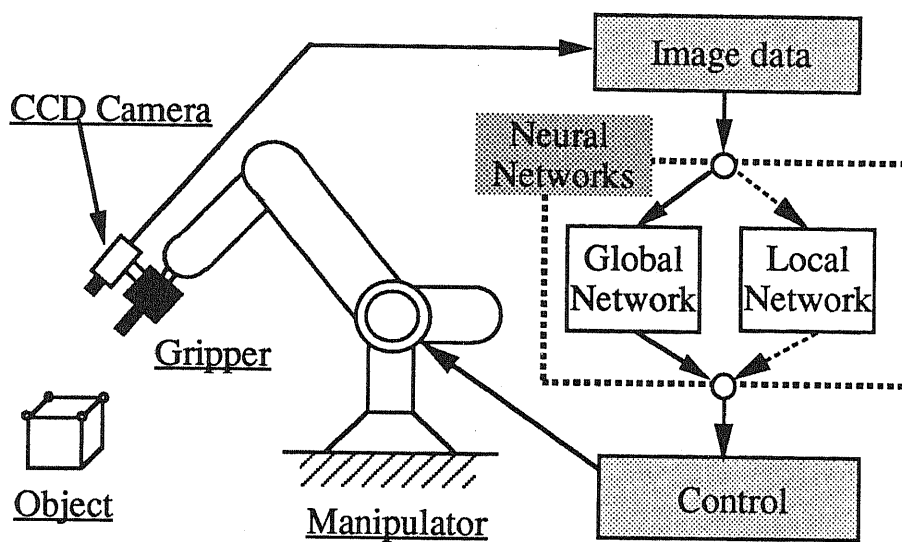


Figure A.21 Visual control system

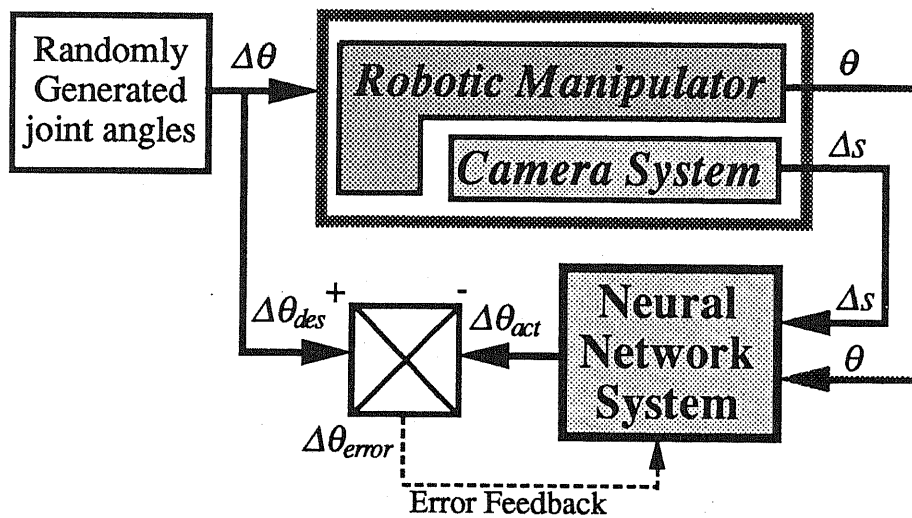


Figure A.22 Learning process

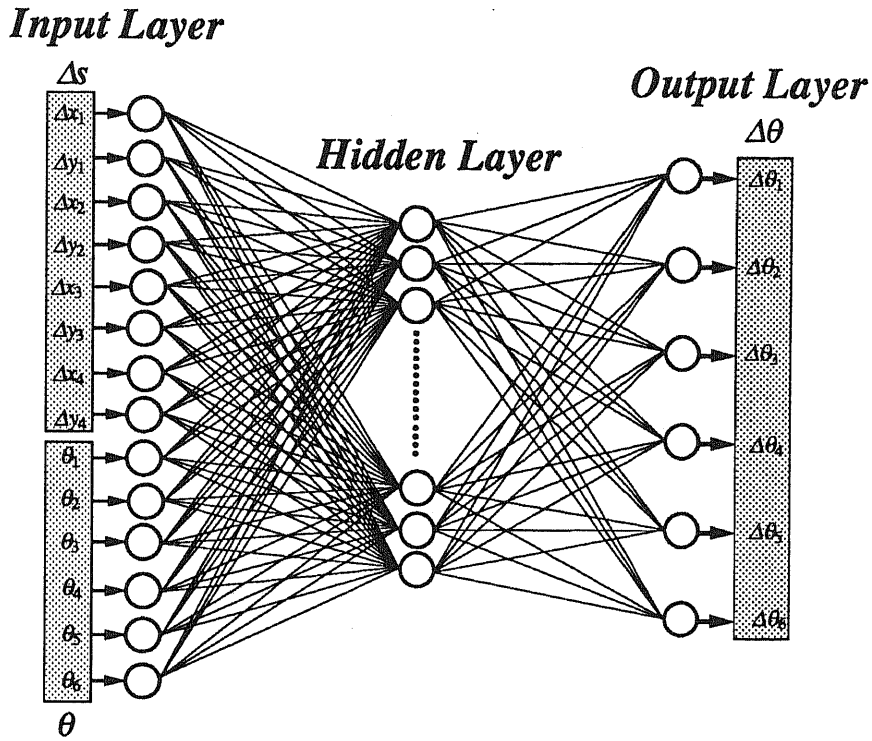


Figure A.23 Neural Network system (Global Network)

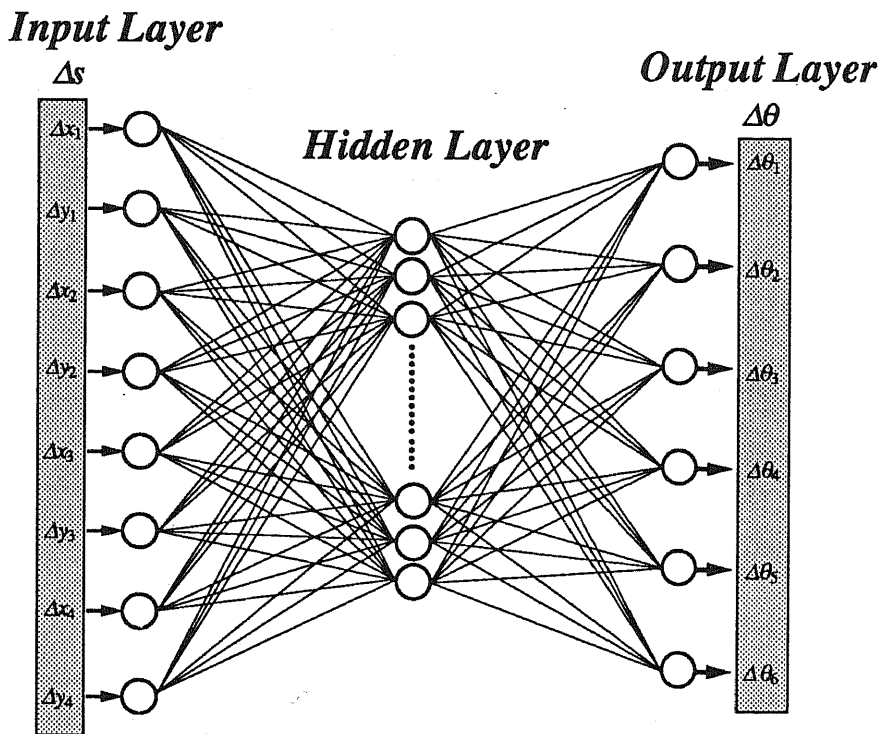


Figure A.24 Neural Network system (Local Network)

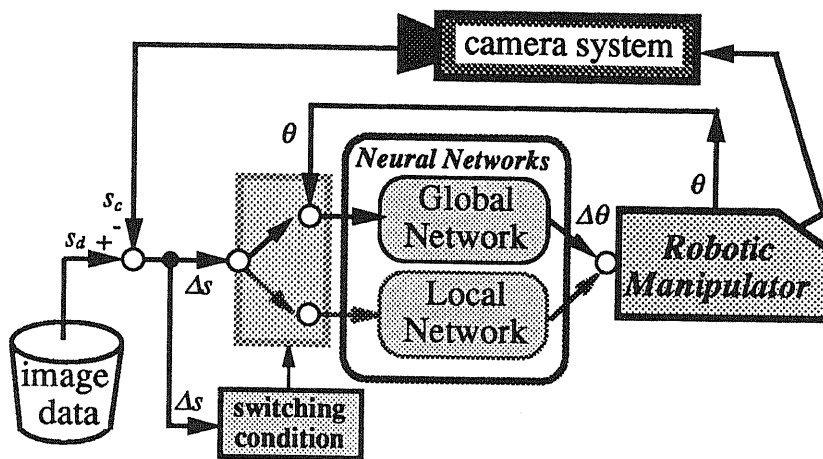


Figure A.25 Execution process

If  $\forall m, n (m \neq n) \alpha_{mn} \leq \alpha_{th}$

Then *Local Neural Network*  
 Otherwise *Global Neural Network*

Here,

$$\alpha_{mn} = \left| \frac{L_{mn}^* - L_{mn}}{L_{mn}^*} \right|, m, n \in \{A, B, C, D\}$$

$\alpha_{th}$  : a threshold value

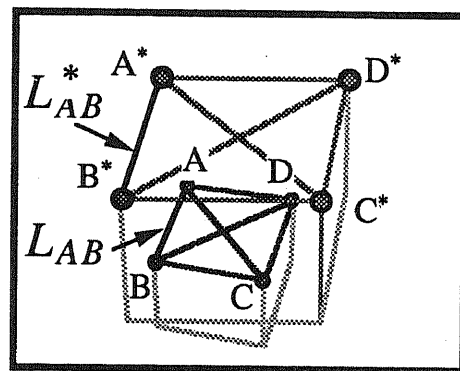


Figure A.26 Change rule

## A.5 シミュレーション

本手法を6自由度マニピュレータに対して適用し、シミュレーションを行った。シミュレータとしてNeuralWorksを用いており、ニューラルネットの初期値等はランダムに設定できる。学習則は、通常のバックプロパゲーションを用いた。

### A.5.1 学習プロセス

カメラ画像の左上の座標値を(0,0)としたとき、目標画像での特徴点4点の座標値を次に示す。

$$\begin{aligned}A^* &= (192, 207), & B^* &= (91, 120) \\C^* &= (293, 120), & D^* &= (192, 274)\end{aligned}$$

カメラ視野の制限により、シミュレーションは、Figure A.27 に示す範囲内について行う。学習データとしては、グローバルネットの場合には各関節角度を1度刻みに、ローカルネットの場合には各関節角度を0.1度刻みにランダムに発生させ、それぞれ60パターン、および100パターンの教師信号を得る。そして、各々のパターンでの関節角度をマニピュレータに入力し、その時得られる画像データ(初期画像)と目標画像との偏差を計算してニューラルネットへの入力とする。ニューラルネットワークへの教師信号は各関節角度である。Table A.5 および Table A.6 に学習信号パターンのうちのいくつかのパターン例をそれぞれ示す。また、本シミュレーションにおいて学習を行った各ニューラルネットワークの構成を Figure A.28 および Figure A.29 にそれぞれ示す。これらのデータを繰り返し教え、約50,000回の学習を行った。

### A.5.2 実行プロセス

実行プロセスでは、まず最初に学習プロセスにおいて60パターンのデータを学習したグローバルニューラルネットワークを用いて、実際にニューラルネットワークからの出力値を制御入力としてマニピュレータを動かす。Figure A.30 は、学習させたパターンに対するシミュレーション結果を示したものであり、目標画像と一致することがわかる。そのときの特徴点4点についての画素エラーを Table A.7 に示す。Table A.7 の上段(初期)は、目標画像と初期画像における特徴点4点の画素の偏差をとったものであり、下段(最終)は、目標画像とニューラルネットからの出力をマニピュ

レータに入力して動かしたときに得られた画像の特徴点4点の画素エラーを表したものである。このデータより最終エラーは  $-4 \sim +4$  画素程度に抑えられた。この結果より、学習したパターンについては、マニピュレータの先端を目標とする位置・姿勢にほぼ移動することができる。

### A.5.3 未学習パターンへの適用

先に学習した 60 パターン以外のデータについて適用した場合についてシミュレーションを行った。データとしては、第 A.3.5 節のシミュレーション3においてエラーの大きかったパターンに対して適用した。そのときのシミュレーション結果を Figure A.31 に示す。Figure A.31(1a)(1b) に示すように目標画像から多少ずれているが、切り換えルールに従ってローカルニューラルネットワークを用いることにより、エラーを減少させることができる。ローカルネットを用いたシミュレーション結果を Figure A.31(1c)(2c) に示す。それ以外の未学習パターンとして学習パターンと同様に Figure A.27 に示した範囲内でランダムに発生させ、そのデータを実行プロセスに従って実行する。そのときの画素エラーを Table A.8 に示す。以上の結果より学習していない任意の位置から目標位置・姿勢に制御できる。

次に、未学習パターンとして対象物が所定の位置から少しずれた場合に対する本手法の有効性の検討をおこなった。そのときの画素エラーを Table A.9 に示す。また、シミュレーション結果を Figure A.32 に示す。

以上のことより、対象物の特徴点4点すべてが検出・認識・識別可能であるという条件の下で、マニピュレータの先端を、視覚情報とニューラルネットワークを用いて目標の位置・姿勢にかなり接近させ、ほぼ一致させることができる。

### A.5.4 ニューラルネットワークに関する考察

ニューラルネットワークの入力として状態角度をいれる場合と入れない場合の比較を行うためにシミュレーションを行った。シミュレーション結果を Table A.10 に示す。ここで、ニューラルネットの構成を次の式で表わす。

$$\Delta\theta = NN_1(\Delta s, \theta) \quad (\text{A.38})$$

$$\Delta\theta = NN_2(\Delta s) \quad (\text{A.39})$$

マニピュレータの位置・姿勢制御入力は、現在の状態（位置）に依存するので、現在の状態角度情報を取入れたほうが正確な制御情報を生成できると考えられる。シミュレーション結果も一致している。ローカルネットワークに関しても、同じことがいえられると思われるが、学習範囲を目標近傍の微小空間としているため特徴点座標偏差と制御関節角空間との関係が線形に近似可能であり、またニューラルネットの入力空間を規格化するため現在の状態を入力するとその影響が大きくなり、状態角度を入力するとかえって悪くなっている。したがって、グローバルネットワークに対しては式(A.38)のタイプを、一方ローカルネットワークに対しては式(A.39)のタイプを用いる。

	Global Network	Local Network
$\theta_1$	$-5.0^\circ \sim 5.0^\circ$	$-1.5^\circ \sim 1.5^\circ$
$\theta_2$	$-30.0^\circ \sim -20.0^\circ$	$-1.5^\circ \sim 1.5^\circ$
$\theta_3$	$15.0^\circ \sim 25.0^\circ$	$-1.5^\circ \sim 1.5^\circ$
$\theta_4$	$-5.0^\circ \sim 5.0^\circ$	$-1.5^\circ \sim 1.5^\circ$
$\theta_5$	$-5.0^\circ \sim 5.0^\circ$	$-1.5^\circ \sim 1.5^\circ$
$\theta_6$	$-5.0^\circ \sim 5.0^\circ$	$-1.5^\circ \sim 1.5^\circ$

( $1.0^\circ$ step)	▼	▼	( $0.1^\circ$ step)
	60patterns	100patterns	
	N.N. $14 \times 20 \times 6$	N.N. $8 \times 30 \times 6$	
	L.T. 50,000	L.T. 50,000	

Figure A.27 Operating range

Table A.5 Training patterns (Global Neural Network)

Trainig No.	Input Patterns [pixel, °]													
	$\Delta x_1$	$\Delta y_1$	$\Delta x_2$	$\Delta y_2$	$\Delta x_3$	$\Delta y_3$	$\Delta x_4$	$\Delta y_4$	$\theta_1$	$\theta_2$	$\theta_3$	$\theta_4$	$\theta_5$	$\theta_6$
1	-8	52	28	70	-20	83	23	102	10.0	50.0	13.0	3.0	15.0	5.0
2	-55	18	-25	44	-74	44	-37	72	13.0	50.0	13.0	3.0	13.0	0.0
3	32	-29	67	7	10	3	51	39	8.0	59.0	9.0	0.0	6.0	-2.0
4	-26	71	6	106	-53	102	-13	137	12.0	58.0	14.0	0.0	6.0	-3.0
5	-1	-59	34	-31	-18	-29	24	-1	10.0	54.0	10.0	5.0	8.0	-1.0
6	41	40	71	66	24	68	60	96	8.0	52.0	11.0	4.0	15.0	-4.0
7	-28	8	6	42	-52	40	-11	75	12.0	58.0	12.0	-3.0	5.0	3.0
8	88	-55	115	-19	66	-27	100	9	5.0	56.0	11.0	-4.0	5.0	-3.0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
60	55	-66	97	-41	44	-30	92	-5	7.0	57.0	5.0	4.0	11.0	5.0

Trainig No.	Output Patterns [°]					
	$\Delta\theta_1$	$\Delta\theta_2$	$\Delta\theta_3$	$\Delta\theta_4$	$\Delta\theta_5$	$\Delta\theta_6$
1	0.0	-20.0	17.0	-3.0	-5.0	-5.0
2	-3.0	-20.0	17.0	-3.0	-3.0	-5.0
3	2.0	-29.0	21.0	0.0	4.0	2.0
4	-2.0	-28.0	16.0	0.0	4.0	3.0
5	0.0	-24.0	20.0	-5.0	2.0	1.0
6	2.0	-22.0	19.0	-4.0	-5.0	4.0
7	-2.0	-28.0	18.0	3.0	5.0	-3.0
8	5.0	-26.0	19.0	4.0	5.0	3.0
⋮	⋮	⋮	⋮	⋮	⋮	⋮
60	3.0	-27.0	25.0	-4.0	-1.0	-5.0

Table A.6 Training patterns (Local Neural Network)

Trainig No.	Input Patterns [pixel, °]								Output Patterns [°]					
	$\Delta x_1$	$\Delta y_1$	$\Delta x_2$	$\Delta y_2$	$\Delta x_3$	$\Delta y_3$	$\Delta x_4$	$\Delta y_4$	$\Delta\theta_1$	$\Delta\theta_2$	$\Delta\theta_3$	$\Delta\theta_4$	$\Delta\theta_5$	$\Delta\theta_6$
1	-39	38	-40	40	-46	37	-45	41	-1.5	-0.4	-0.2	1.0	-1.0	0.7
2	-19	21	-19	22	-22	22	-21	22	-0.7	-0.1	-1.0	1.2	0.7	1.4
3	-11	-14	-12	-13	-12	-16	-12	-15	-0.4	0.4	0.7	1.0	-1.0	-1.2
4	31	32	34	32	33	37	37	36	1.2	-1.2	0.4	1.0	-0.7	-1.3
5	21	6	22	5	22	8	25	7	0.8	-0.7	-0.3	-1.2	1.2	0.9
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
100	1	3	4	2	3	6	7	4	0.1	-0.5	-0.1	-0.1	0.7	-1.2

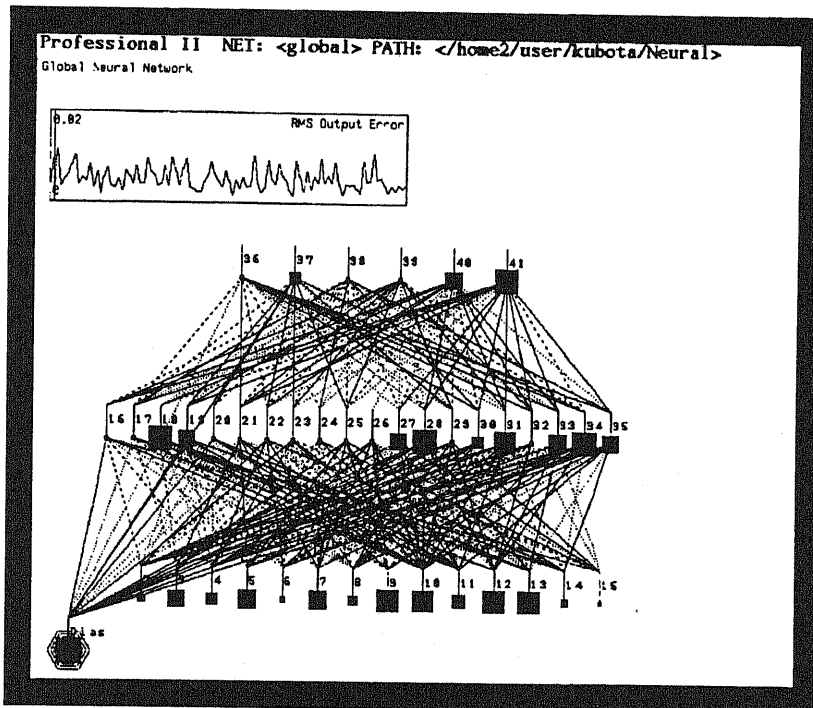


Figure A.28 Global Network

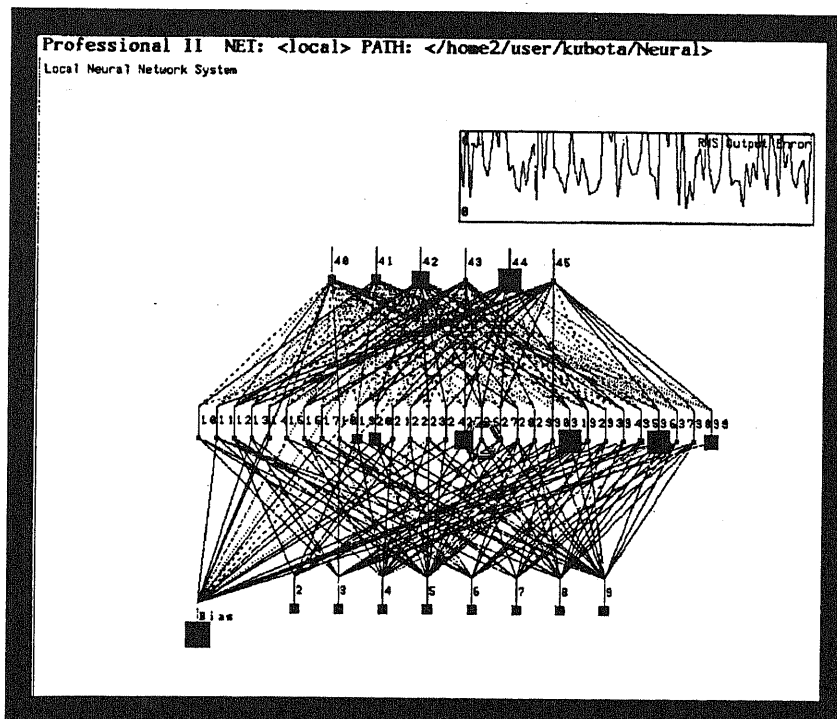


Figure A.29 Local Network



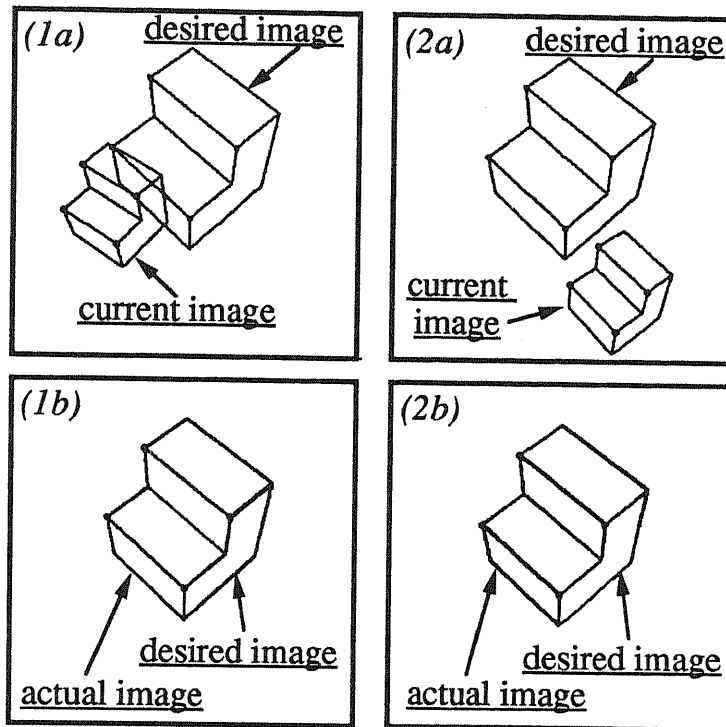


Figure A.30 Simulation results (1) [learned patterns]

—— Global Network ——

Table A.7 Pixel error (learned patterns)

Errors	$\Delta x_1$	$\Delta y_1$	$\Delta x_2$	$\Delta y_2$	$\Delta x_3$	$\Delta y_3$	$\Delta x_4$	$\Delta y_4$
Initial	-81	31	-52	71	-111	57	-73	99
Final	0	-1	0	-1	-1	-1	0	-1
Initial	58	116	94	142	42	151	84	177
Final	-1	1	-1	2	-2	1	-1	1
Initial	-51	54	-20	86	-76	82	-37	116
Final	1	-3	1	-2	1	-3	2	-1

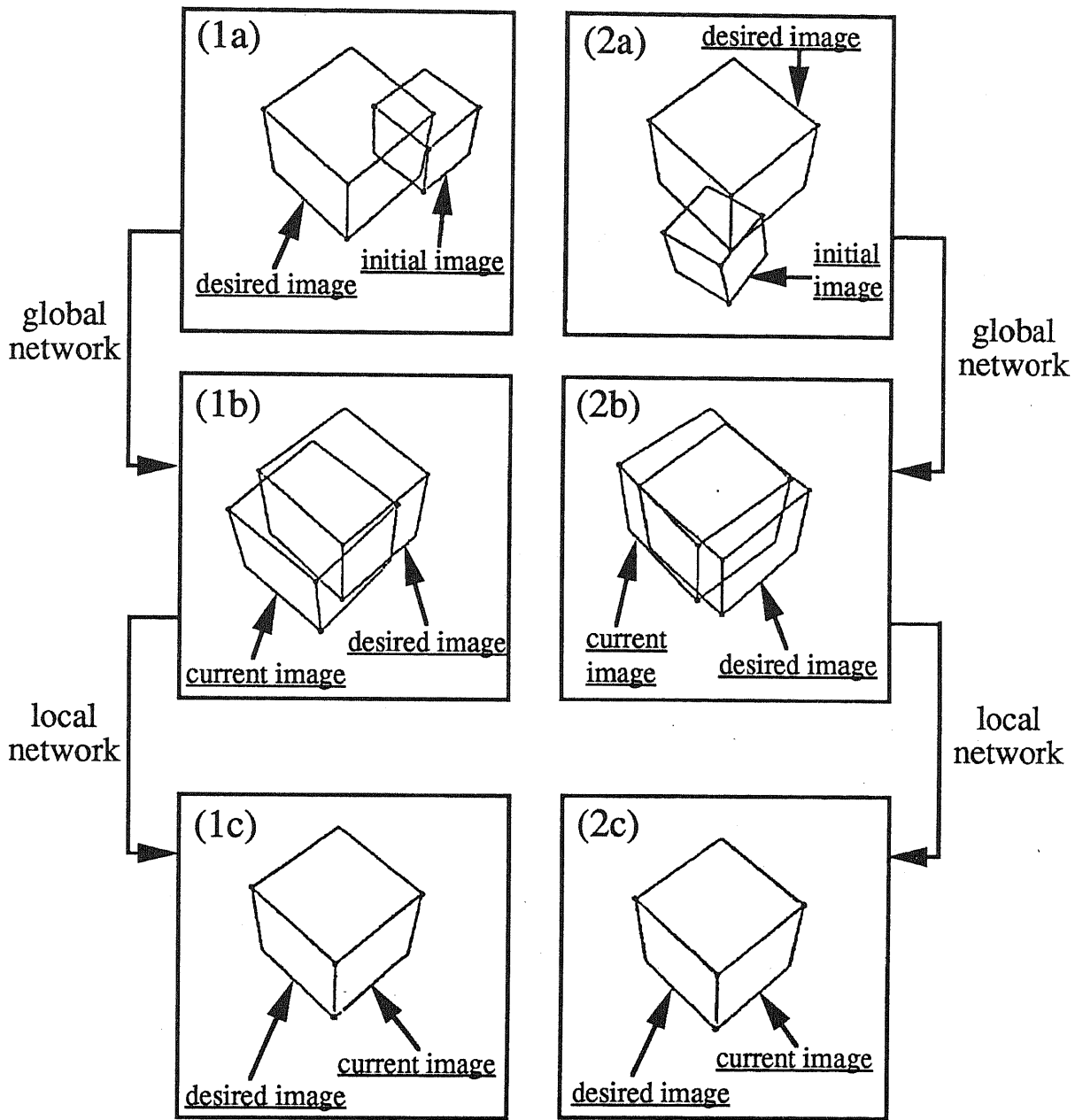


Figure A.31 Simulation results (2)

Table A.8 Pixel error (2) (not learned pattern)

Error	$\Delta x_1$	$\Delta y_1$	$\Delta x_2$	$\Delta y_2$	$\Delta x_3$	$\Delta y_3$	$\Delta x_4$	$\Delta y_4$
Initial	-1	64	-12	86	-64	110	20	135
Feedback	-30	-17	-30	-17	-22	-12	-25	-25
Final	-3	0	-2	2	-1	3	-1	-1
Initial	91	-60	96	-45	54	-9	130	-7
Feedback	-25	38	-32	45	36	37	-36	47
Final	-3	-1	-1	2	1	2	-1	2
Initial	18	-49	13	-33	-33	-1	48	13
Feedback	0	-11	-1	-14	-2	-14	-3	-11
Final	0	-4	0	-6	-1	-7	-3	-4

Table A.9 Pixel error (3) (not learned pattern)

Error	$\Delta x_1$	$\Delta y_1$	$\Delta x_2$	$\Delta y_2$	$\Delta x_3$	$\Delta y_3$	$\Delta x_4$	$\Delta y_4$
Initial	-29	98	1	130	-55	127	-17	160
Global Net	2	25	-2	18	6	23	2	15
Local Net	-3	5	-3	-1	0	5	1	-2
Initial	54	30	96	53	43	68	92	92
Global Net	-10	-22	-4	-15	-14	-19	-6	-12
Local Net	0	-4	4	0	-2	-2	4	2
Initial	-4	89	26	122	-30	118	8	152
Global Net	11	27	4	23	13	23	6	20
Local Net	-8	0	-8	-5	-6	1	-5	-5

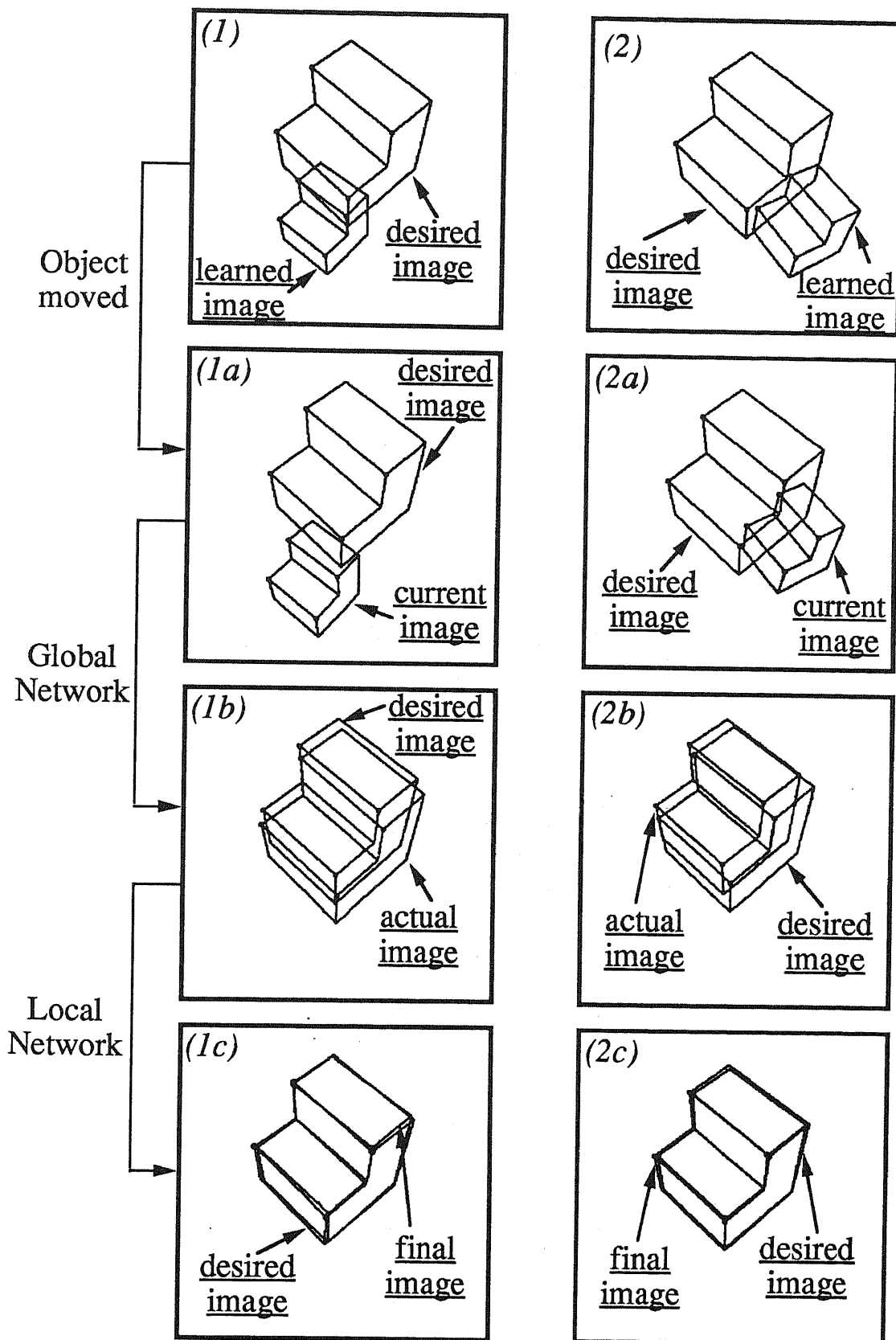


Figure A.32 Simulation results (3) [not learned pattern]

Table A.10 Simulation results (pixel error)

———— Global Network ————

Hidden Layer : 20unit  
 Learning patterns : 60  
 Learning times : 50,000

Try No.	Initial							
	$\Delta X_1$	$\Delta Y_1$	$\Delta X_2$	$\Delta Y_2$	$\Delta X_3$	$\Delta Y_3$	$\Delta X_4$	$\Delta Y_4$
1	-3	48	33	67	-15	79	28	99
2	-26	63	10	80	-38	94	5	112
3	-31	45	6	64	-44	76	0	97
4	57	44	92	66	45	77	86	100
Try No.	Final NN( $\Delta s$ )							
	$\Delta X_1$	$\Delta Y_1$	$\Delta X_2$	$\Delta Y_2$	$\Delta X_3$	$\Delta Y_3$	$\Delta X_4$	$\Delta Y_4$
1	-19	12	-21	12	-21	10	-21	10
2	38	35	38	27	45	36	45	28
3	21	43	26	35	29	46	35	39
4	17	72	18	67	17	76	19	72
Try No.	Final NN( $\Delta s, \theta$ )							
	$\Delta X_1$	$\Delta Y_1$	$\Delta X_2$	$\Delta Y_2$	$\Delta X_3$	$\Delta Y_3$	$\Delta X_4$	$\Delta Y_4$
1	0	1	-1	3	-2	1	-1	3
2	4	-4	4	-7	6	-5	6	-8
3	1	-29	4	-29	2	-29	5	-29
4	1	12	4	16	-3	16	2	20

## A.6 実験による検討

実験システムを Figure A.33 に示す。本実験において用いたマニピュレータは5自由度を有し、あらかじめオフラインで学習済みのニューラルネットワークを用いて制御入力生成した。対象物の特徴点4点の認識は、画像処理に負担をかけないために実験では色情報によって識別している。本実験により提案する手法の有効性を確認した。

### A.6.1 実験システム

本手法を5自由度マニピュレータ (*MOVEMASTER-EX*) に対して適用した。実験システムの外観を Figure A.34 に示す。マニピュレータの先端に小型のカラーCCDカメラ (*XC711RR*) 1台が取り付けられており、画像情報はイメージメモリボード (*FDM98RGB*) を通して解像度  $256 \times 256$ 、輝度レベル *6bit* のカラー画像データに変換され、ホストコンピュータのVRAM上へ一画像当たり約  $0.03[\text{sec}]$  以内にDMA転送される。対象物の特徴点の認識は、画像処理に負担をかけないように色情報によって識別している。ニューラルネットワークシミュレータとしてトランスピュータ5個からなる *TRP-NEURO SIMULATOR* (網) を用いており、ニューラルネットの初期値等はランダムに設定できる。Figure A.35 を示す。学習則は、通常のバックプロパゲーションを用いた。

### A.6.2 ニューラルネットの学習

シミュレーションに用いたニューラルネットワークの構成及びマニピュレータの可動範囲を Figure A.36、Figure A.37 にそれぞれ示す。学習データとしては、グローバルネットにおいては60パターン、ローカルネットにおいては100パターンの教師信号を用意し、約2,000回で収束したが精度を上げるために約50,000回の学習を行った。

### A.6.3 実験結果

実行プロセスでは、まず最初に学習プロセスにおいて学習したグローバルニューラルネットワークを用いて、実際にニューラルネットワークからの出力値を制御入力としてマニピュレータを動かす。そのときの特徴点4点についての画素エラーを Table A.11 に示す。このデータより学習パターンについては、特徴点(4点)の画素エラーは  $-5 \sim +5$  画素程度に抑えられた。

学習したパターン以外のデータについて Figure A.37 に示した範囲以内でランダムに発生させ実験を行った。ここでは学習した空間内の任意の位置から目標位置・姿勢に制御を行っている。目標画像に完全に一致はしていないが、大まかに位置・姿勢の制御を行うことができる。

次に、未学習パターンとして Figure A.38 に示すように対象物が所定の位置から少しずれた場合に対する本手法の有効性の検討をおこなった。そのときの画素エラーを Table A.12 に示す。ローカルニューラルネットワークを繰り返し用いることによりエラーを減少させることができる。

以上より、マニピュレータの先端は、視覚情報とニューラルネットワークを用いて目標の位置・姿勢にかなり接近し、ほぼ一致させることができる。

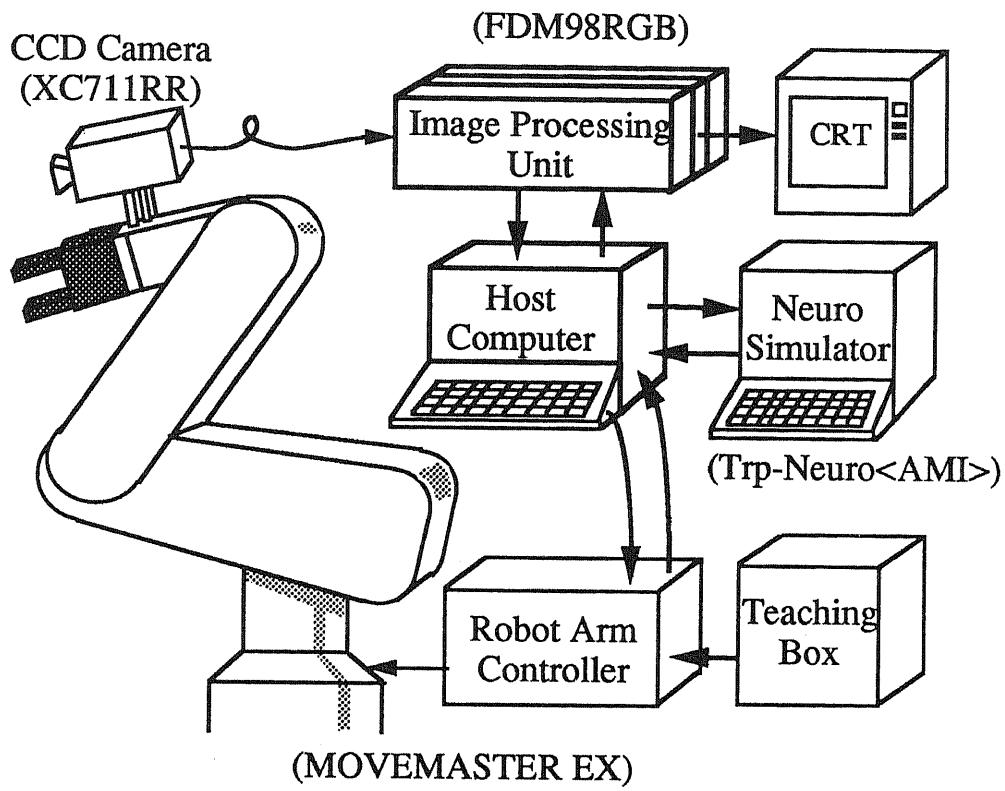


Figure A.33 Configuration of experimental system

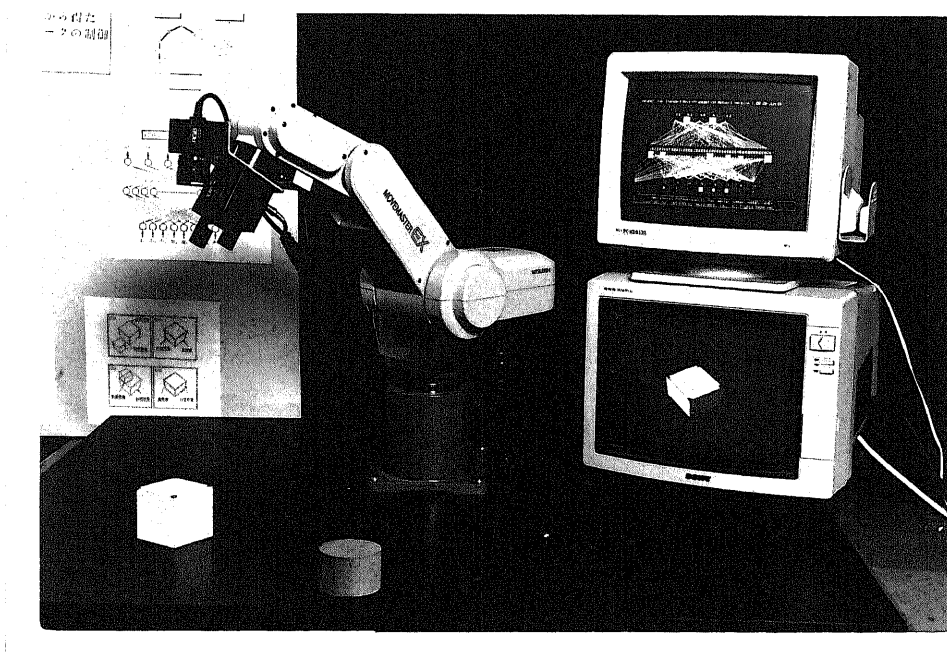
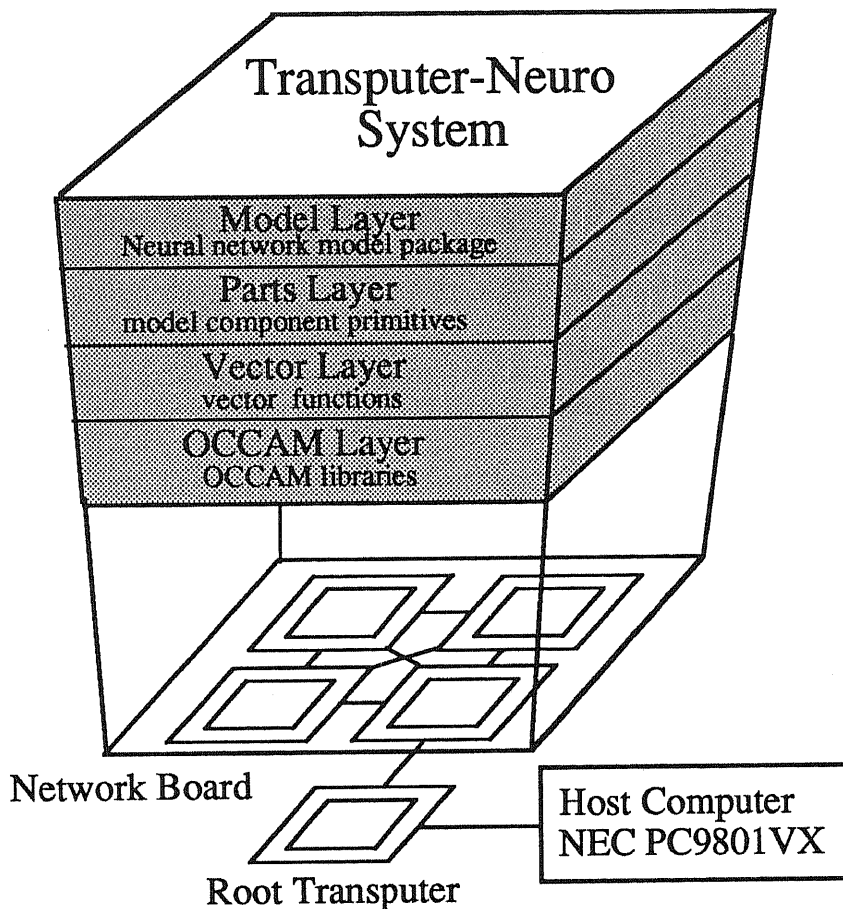


Figure A.34 Experimental system





Model : Backpropagation

Network configuration :   Input Layer   13unit  
                                   Hidden Layer  30unit  
                                   Output Layer  5unit

Learning patterns : 60

Learning times    : 50,000

Learning time     : 10 [hr]

Figure A.35 Trp-neuro simulator (AMI)

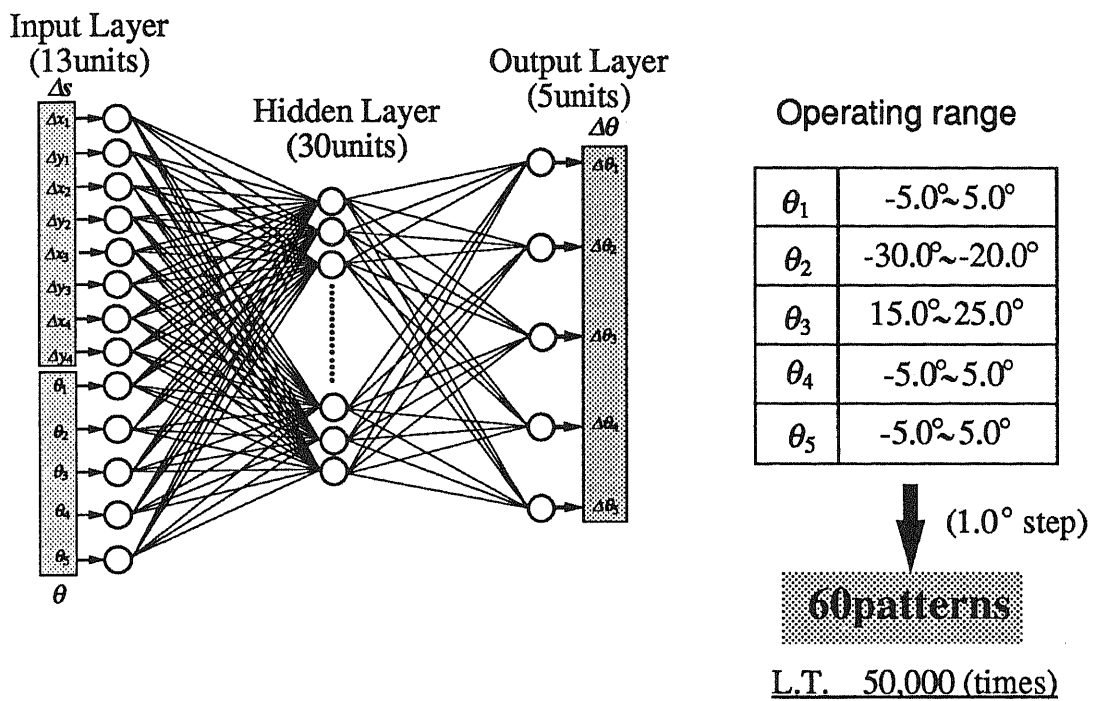


Figure A.36 Neural Network (Global Network)

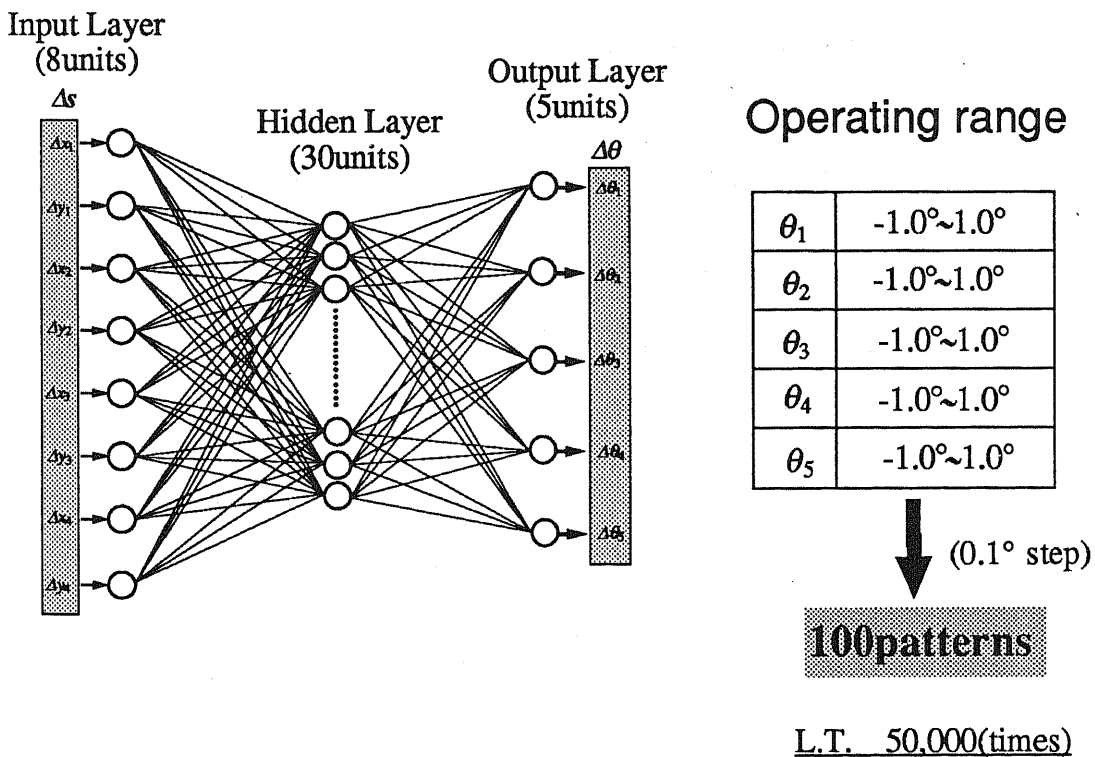


Figure A.37 Neural Network (Local Network)

Table A.11 Experimental results [learned patterns]

———— Global Network ————

Pixel errors	$\Delta x_1$	$\Delta y_1$	$\Delta x_2$	$\Delta y_2$	$\Delta x_3$	$\Delta y_3$	$\Delta x_4$	$\Delta y_4$
Initial	42	-72	-18	-9	41	-9	-20	-71
Final	0	0	-2	-3	-2	0	1	-2
Initial	91	21	37	97	100	87	28	34
Final	4	-3	4	-5	4	-3	4	-5
Initial	3	-3	-59	60	1	58	-60	-3
Final	0	3	1	4	1	3	0	4

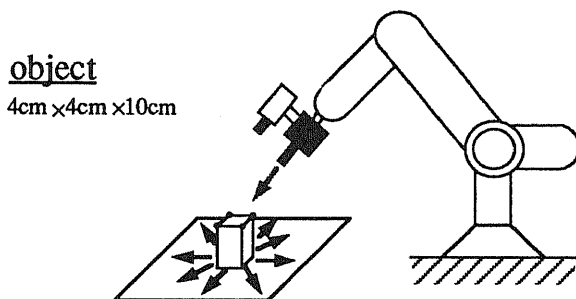


Figure A.38 Experiment for not learned pattern

Table A.12 Experimental results [not learned patterns]

Pixel errors	$\Delta x_1$	$\Delta y_1$	$\Delta x_2$	$\Delta y_2$	$\Delta x_3$	$\Delta y_3$	$\Delta x_4$	$\Delta y_4$
Initial	3	-69	-53	-14	1	-12	-54	-71
Global Net	-11	-12	-9	-9	-8	-11	-11	-11
Local Net	0	-6	-2	0	1	-3	-2	-3
Initial	37	-63	-17	-9	37	-9	-17	-63
Global Net	12	-12	16	-12	13	-14	16	11
Local Net	3	-3	4	-3	4	-3	4	-2
Initial	0	-42	-53	10	1	12	-54	-44
Global Net	-21	12	-11	7	-17	6	-14	13
Local Net	0	6	1	0	-2	3	3	4

## A.7 ビジュアル制御法に対する考察

### A.7.1 最小2乗推定法との比較検討

#### (1) 線形近似法

画像データとマニピュレータの関節角空間との写像関係を次のように考える。

$\theta \in R^6$  : 関節角  
 $P \in R^6$  : マニピュレータ先端の位置・姿勢  
 $x \in R^8$  : 特徴点4点の画像上での座標

$$\begin{cases} P = f(\theta) \\ x = g(P) \\ \Delta\theta = NN(x_d - x) = NN(\Delta x) \end{cases} \quad (\text{A.40})$$

目標位置・姿勢  $(x_d, \theta_d, P_d)$  において

$$\begin{cases} P_d = f(\theta_d) \\ x_d = g(P_d) \end{cases} \quad (\text{A.41})$$

式(A.40)より、

$$x = g(f(\theta)) \quad (\text{A.42})$$

$x_d$ 、 $\theta_d$ 、 $P_d$  の近傍において

$$x_d + \Delta x \approx g(f(\theta_d)) + \frac{\partial}{\partial \theta} g(f(\theta)) \Big|_{\theta=\theta_d} \cdot \Delta \theta \quad (\text{A.43})$$

式(A.41)、式(A.43)より、

$$\Delta x = \frac{\partial}{\partial \theta} g(f(\theta)) \Big|_{\theta=\theta_d} \Delta \theta \quad (\text{A.44})$$

いま、 $N \in R^{6 \times 8}$  とすると式(A.44)を式(A.40)に代入して

$$\Delta\theta = N\left(\frac{\partial}{\partial\theta}g(f(\theta))\right)_{\theta=\theta_d}\Delta\theta \quad (\text{A.45})$$

$$\implies N\frac{\partial}{\partial\theta}(g(f(\theta)))|_{\theta=\theta_d} = I_6 \quad (\text{A.46})$$

ひとつの解として

$$N = \left[\left(\frac{\partial}{\partial\theta}g(f(\theta))\right)^T \frac{\partial}{\partial\theta}g(f(\theta))\right]^{-1} \left(\frac{\partial}{\partial\theta}g(f(\theta))\right)^T |_{\theta=\theta_d} \quad (\text{A.47})$$

となる。これは、 $\Delta\theta$ 、 $\Delta x$  が与えられたときの解になる。 $x_d$ 、 $\theta_d$ 、 $P_d$  の近傍において線形モデルとみなすと次の式が得られる。

$$\begin{cases} \Delta\theta_1 = \tilde{c}_{11}\Delta x_1 + \tilde{c}_{12}\Delta x_2 + \cdots + \tilde{c}_{18}\Delta x_8 \\ \Delta\theta_2 = \tilde{c}_{21}\Delta x_1 + \tilde{c}_{22}\Delta x_2 + \cdots + \tilde{c}_{28}\Delta x_8 \\ \vdots \\ \vdots \\ \Delta\theta_6 = \tilde{c}_{61}\Delta x_1 + \tilde{c}_{62}\Delta x_2 + \cdots + \tilde{c}_{68}\Delta x_8 \end{cases} \quad (\text{A.48})$$

したがって、

$$\Delta\theta = \tilde{c} \Delta x \quad (\text{A.49})$$

となる。ここで、 $\tilde{c}$  を推定する。

いま、 $\tilde{c}$  を推定するためにデータ  $(\Delta x^n, \Delta\theta^n)$  ( $n = 1, 2, \dots, 8$ ) の8組を用意する。

$$\Theta_i = \begin{bmatrix} \theta_i^1 \\ \theta_i^2 \\ \vdots \\ \theta_i^8 \end{bmatrix} \in R^{8 \times 1} \quad (\text{A.50})$$

$$X_i = \begin{bmatrix} \Delta x_1^1 & \Delta x_2^1 & \cdots & \Delta x_8^1 \\ \Delta x_1^2 & \Delta x_2^2 & \cdots & \Delta x_8^2 \\ \vdots & \vdots & \ddots & \vdots \\ \Delta x_1^8 & \Delta x_2^8 & \cdots & \Delta x_8^8 \end{bmatrix} \in R^{8 \times 8} \quad (\text{A.51})$$

$$\tilde{C}_i = \begin{bmatrix} \tilde{C}_i^1 \\ \tilde{C}_i^2 \\ \vdots \\ \tilde{C}_i^8 \end{bmatrix} \in R^{8 \times 1} \quad (\text{A.52})$$

とすると

$$\Theta = X \cdot \tilde{C} \quad \Theta = [\Theta_1, \Theta_2, \dots, \Theta_6] \quad \tilde{C} = [\tilde{C}_1, \tilde{C}_2, \dots, \tilde{C}_6] \quad (\text{A.53})$$

式(A.53)において、 $X$  が *nonsingular* ( $|X| \neq 0$ ) であるならば、 $\tilde{C}$  は次のように推定される。

$$\tilde{C} = X^{-1} \cdot \Theta \quad (\text{A.54})$$

## (2) 最小2乗推定法

ここでは最小2乗法によるパラメータ推定を行う。マニピュレータの各関節角をある微小範囲内で動かした場合画像上での特徴点4点の座標空間と各関節角空間との関係を線形モデルに置き換えると次のように表現できる。

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_6 \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{18} \\ c_{21} & c_{22} & \cdots & c_{28} \\ \vdots & \vdots & \ddots & \vdots \\ c_{61} & c_{62} & \cdots & c_{68} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_8 \end{bmatrix} \quad (\text{A.55})$$

これより、未知パラメータ  $C$  を最小2乗推定する。

いま、マニピュレータの第  $i$  番目の関節  $\theta_i$  に注目すると

$$\theta_i^t = X^{tT} c_i + r^t$$

$$c_i^T = [c_{i1}, c_{i2}, \dots, c_{i8}] \quad (\text{A.56})$$

$$x^T = [x_1, x_2, \dots, x_8]$$

$$t = 1, 2, \dots, N(\text{データ数})$$

次の線形回帰モデルが得られる。

$$\Theta = XC + r \quad (\text{A.57})$$

最小2乗法によるパラメータ推定は、次の評価関数  $J$  を最小にする  $C$  を求めることである。

$$\theta_i^t = X^{tT} c_i + r^t \quad (\text{A.58})$$

$$J = \sum_{t=1}^N r_t^2 \quad (\text{A.59})$$

$$= (\Theta - XC)^T (\Theta - XC) \quad (\text{A.60})$$

$J$  を最小にする条件は、

$$\frac{\partial J}{\partial C} = 0 \quad (\text{A.61})$$

である。したがって、次のようになる。

$$\hat{C}_i = (X^T X)^{-1} X^T \Theta \quad (\text{A.62})$$

$$= \left( \sum_{t=1}^N x^t x^{tT} \right)^{-1} \left( \sum_{t=1}^N x^t \theta_i^t \right) \quad (\text{A.63})$$

以上により第  $i$  関節についての未知パラメータが推定できる。逐次計算方式は次のようになる。

$$P_N = \left( \sum_{t=1}^N x^t x^{tT} \right)^{-1} \quad (\text{A.64})$$

とすると

$$P_N^{-1} = \sum_{t=1}^{N-1} x^t x^{tT} + x_N x_N^T \quad (\text{A.65})$$

$$= P_{N-1}^{-1} + x_N x_N^T \quad (\text{A.66})$$

したがって、

$$\hat{C}_N = \hat{C}_{N-1} + P_N x_N (\theta_N - x_N^T \hat{C}_{N-1}) \quad (\text{A.67})$$

$$P_N^{-1} = P_{N-1}^{-1} + x_N x_N^T \quad (\text{A.68})$$

逆行列を用いずに示すと、

$$\hat{C}_N = \hat{C}_{N-1} + \frac{P_{N-1}x_N}{1 + x_N^T P_{N-1} x_N} (\theta_N - x_N^T \hat{C}_{N-1}) \quad (\text{A.69})$$

$$P_N = P_{N-1} - \frac{P_{N-1}x_N x_N^T P_{N-1}}{1 + x_N^T P_{N-1} x_N} \quad (\text{A.70})$$

式(A.69)が逐次計算アルゴリズムある。ここで、初期値は次のように設定される。

$$\hat{C}_0 = \text{任意 (通常 0)} \quad (\text{A.71})$$

$$P_0 = \alpha I \quad (\alpha = 10^4 \sim 10^5) \quad (\text{A.72})$$

### (3) シミュレーション結果

ここでは、目標位置の近傍を学習したローカルネットワークと最小2乗法との比較をおこなった。そのシミュレーション結果を Table A.13 および Table A.14 に示す。ニューラルネットを用いた手法と最小2乗法との差はあまりみられないが、ニューラルネット手法の方が誤差が若干小さくなっている。また、最小2乗法では、全体の誤差の2乗和を最小とするので学習パターンに対しても誤差が生じている。

#### A.7.2 ニューラルネットの補間能力

ニューラルネットのメリットはその汎化能力にあるが、これは学習パターンの与え方に依存し、また過剰学習の問題もある。したがって、十分な汎化能力を得るために必要な学習パターンをどのように生成すればよいかは重要な問題であり、現在のところ問題に応じて試行錯誤的に作られている。また、ニューラルネットの学習を過剰に行うと、学習パターンに対して過度に発火し、連続非線形関数のモデリングが困難になる。これはニューラルネットが、入力空間から出力空間への点から点への対応関係を学習するためである。したがって、入力空間と出力空間のマッピングを点間の対応ではなく、線分あるいは面の対応関係を学習する方法 [川端 90] を導入すれば学習点以外の点に関しても内挿できると考えられる。



Table A.13 Pixel error by Neural Network method

Error	$\Delta x_1$	$\Delta y_1$	$\Delta x_2$	$\Delta y_2$	$\Delta x_3$	$\Delta y_3$	$\Delta x_4$	$\Delta y_4$
Initial	-41	14	-38	16	-29	28	-26	7
Final	-1	2	-1	0	-1	0	-2	1
Initial	-19	-12	-32	-13	-41	-23	-40	1
Final	-2	2	-4	-1	-5	1	-4	-1
Initial	28	-35	27	-38	20	-45	18	-30
Final	2	-3	1	-4	2	-7	3	-5

Table A.14 Pixel error by Least Square Estimation method

Error	$\Delta x_1$	$\Delta y_1$	$\Delta x_2$	$\Delta y_2$	$\Delta x_3$	$\Delta y_3$	$\Delta x_4$	$\Delta y_4$
Initial	-41	14	-38	16	-29	28	-26	7
Final	-7	2	-5	3	-4	5	-8	2
Initial	-19	-12	-32	-13	-41	-23	-40	1
Final	-5	4	-9	-6	-12	7	-5	-2
Initial	28	-35	27	-38	20	-45	18	-30
Final	4	-10	6	-6	4	-11	6	-9

### A.7.3 ビジュアルサーボについて

ここでは、提案する手法のビジュアルサーボ系への適用可能性について考察する。通常の視覚サーボ系では観測量は画像であり制御量は手先と作業対象物との間の相対的な位置と姿勢である。しかし、3次元的な位置と姿勢は画像入力にともなうノイズと量子誤差が原因で精度よく求められない。さらにカメラが運動するので物体を見失う可能性がある。

そこでローカルネットワークを用いて目標追従のシミュレーションを行った。動的視覚サーボシステムを Figure A.39 に示す。シミュレーション結果を Figure A.40、Figure A.41 に示す。また、画素エラーを Table A.15 に示す。この結果から誤差は若干あるがマニピュレータの手先が目標に追従していると思われる。しかしながら、本手法では目標とする物体の位置情報に関しては画像上の特徴点の座標値から陽ではないが用いているのに対し、物体の運動情報に関しては全く考慮されていない。したがって、本手法によって収束する保証がない。

このようなビジュアルサーボ系に対しては、物体の運動の予測を行い、画像処理にともなう時間遅れを補償することが必要である。たとえば、動画像データあるいはオプティカルフローの情報を取り入れることによって、作業対象物の運動に関する知識を予測し、また画像処理のための時間遅れを補償することによって、マニピュレータの運動性能が向上できると考えられる。したがって、運動情報を考慮した制御メカニズムを本手法に取り込む必要がある。

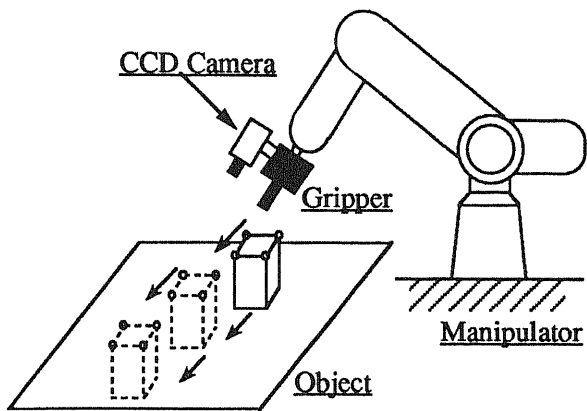


Figure A.39 Dynamic servo system

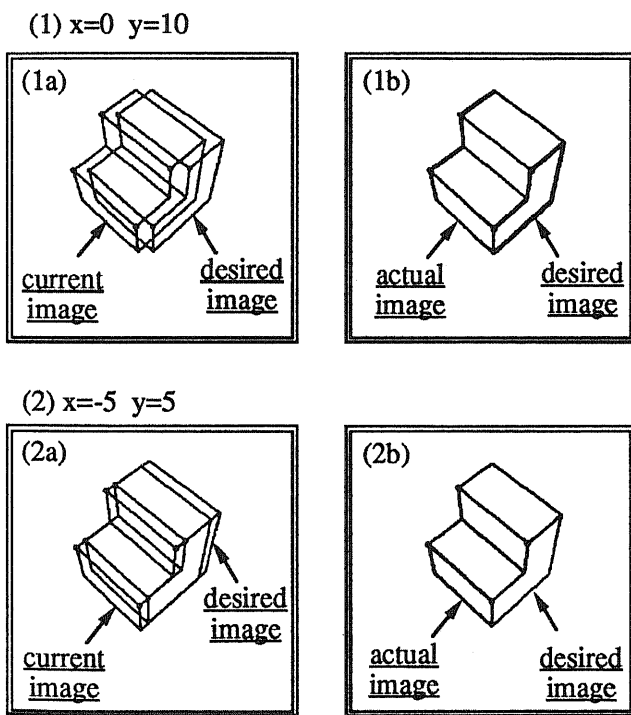


Figure A.40 Simulation results 1

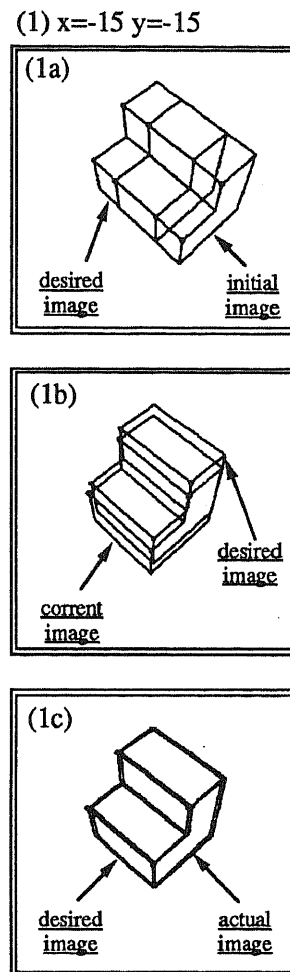


Figure A.41 Simulation result 2

Table A.15 Simulation results

$$\text{Evaluation value} = \frac{1}{4} \sum_i \sqrt{\Delta x_i^2 + \Delta y_i^2}$$

[Case1]

Moving distance		Initial (pixel error)								Final (pixel error)								Evaluation value
X	Y	$\Delta x_1$	$\Delta y_1$	$\Delta x_2$	$\Delta y_2$	$\Delta x_3$	$\Delta y_3$	$\Delta x_4$	$\Delta y_4$	$\Delta x_1$	$\Delta y_1$	$\Delta x_2$	$\Delta y_2$	$\Delta x_3$	$\Delta y_3$	$\Delta x_4$	$\Delta y_4$	
5	5	-11	-10	-10	-9	-14	-10	-11	-9	-3	1	-3	2	-4	1	-3	1	3.51
5	5	-15	-9	-13	-7	-17	-9	-14	-7	0	-4	0	-4	-1	-5	0	-4	4.27
5	5	-12	-15	-10	-13	-14	-15	-11	-13	0	1	-1	2	-1	0	0	-1	1.31
5	5	-12	-9	-11	-7	-14	-10	-11	-8	0	-2	1	-1	0	-3	1	-2	2.16
5	5	-11	-12	-10	-10	-13	-13	-10	-11	0	0	1	1	-1	-1	1	1	1.06
5	5	-11	-10	-9	-8	-14	-11	-10	-8	0	-2	0	-1	-1	-4	0	-3	2.53

[Case2]

Moving distance		Initial (pixel error)								Final (pixel error)								Evaluation value
X	Y	$\Delta x_1$	$\Delta y_1$	$\Delta x_2$	$\Delta y_2$	$\Delta x_3$	$\Delta y_3$	$\Delta x_4$	$\Delta y_4$	$\Delta x_1$	$\Delta y_1$	$\Delta x_2$	$\Delta y_2$	$\Delta x_3$	$\Delta y_3$	$\Delta x_4$	$\Delta y_4$	
0	10	-24	0	-22	0	-26	0	-23	0	1	3	2	2	2	2	3	2	3.11
0	10	-22	3	-21	2	-24	2	-21	2	1	2	2	1	2	2	3	1	2.62
0	10	-22	2	-20	1	-24	1	-20	1	2	2	4	1	3	2	5	1	3.91
0	10	-21	2	-19	1	-23	1	-19	0	2	2	4	1	2	2	4	0	3.44
0	10	-22	2	-19	1	-24	1	-19	0	3	3	5	1	4	2	6	0	4.95
0	10	-20	2	-17	1	-22	1	-18	0	1	2	4	0	2	0	4	1	3.09

[Case3]

Moving distance		Initial (pixel error)								Final (pixel error)								Evaluation value
X	Y	$\Delta x_1$	$\Delta y_1$	$\Delta x_2$	$\Delta y_2$	$\Delta x_3$	$\Delta y_3$	$\Delta x_4$	$\Delta y_4$	$\Delta x_1$	$\Delta y_1$	$\Delta x_2$	$\Delta y_2$	$\Delta x_3$	$\Delta y_3$	$\Delta x_4$	$\Delta y_4$	
-15	-15	37	33	31	28	41	32	35	27	-3	17	-3	14	-3	19	-1	16	16.71
0	0	-3	17	-3	14	-3	19	-1	16	-3	2	-5	1	-4	2	-4	1	4.32
-15	-15	33	35	27	30	37	35	31	29	0	3	-1	1	1	6	1	2	3.18
-15	-15	33	35	27	30	37	35	31	29	1	10	-1	9	1	14	0	10	10.79
0	0	1	10	-1	9	1	14	0	10	-3	0	-7	1	-5	3	-8	1	5.99
-15	-15	34	32	26	29	37	35	29	30	3	-1	0	-3	3	3	1	-1	2.95
-15	-15	40	32	33	26	45	36	38	28	6	7	2	6	6	12	3	8	9.38
0	0	6	7	2	6	6	12	3	8	-1	2	-5	2	-3	7	-5	4	5.41
-15	-15	36	34	28	31	39	40	31	33	7	2	3	0	8	8	5	3	6.86

#### A.7.4 冗長マニピュレータについて

自律型ロボットにおいては、環境の変化に対応する柔軟性や即応性が要求されるので、マニピュレータに対して冗長性をもたせることが必要である。冗長マニピュレータとは、Figure A.42 に示すように多自由度を有するマニピュレータでマニピュレータの先端の位置・姿勢に対して、各関節角度が一对多であるものをいう。冗長マニピュレータは、その冗長性を利用して特異点や障害物回避・対象物の裏側へのまわりこみなどの動作が可能となる。一般にロボットアームはある姿勢においてその手先の位置・姿勢を任意の方向に動かせなくなることがある。このような姿勢は特異点と呼ばれている。特異点はそれ自体を通過することが望ましくないだけでなく、その近傍においても関節角度が過大になることがあるので、それに近づくことも望ましくない。そこで、冗長性を利用して特異点からの距離を表す可操作度がつねに大きくなるように制御すれば特異点を回避することができる。

このように冗長マニピュレータは有用であるが、逆に冗長性があるがゆえに例えば、マニピュレータの先端位置・姿勢からマニピュレータの各関節角度を計算する逆キネマティクスを解く場合に複数の解があり一意に定まらず、また自由度が多くなると演算処理時間も増大する。これまでに提案された制御手法では演算時間がかかるなどリアルタイム制御には不向きである。このような問題を解決するひとつの方法として神経回路網を用いて処理することにより並列処理が可能で応答性が良くなることが期待できる。

ニューラルネットワークを用いて、教えるパターンをうまく選択するならば冗長性を排除することが可能である。これは、入出力が一对多となる関係に対して、一对一となるような写像を選択的に学習によって作り上げることを意味する。本手法の有効性を検証するために、冗長性のある平面5自由度のマニピュレータを対象としてシミュレーションを行なった。Figure A.43 は学習パターンに対するシミュレーション結果を、Figure A.44 は未学習パターンに対するシミュレーション結果をそれぞれ示している。これらより、本手法は冗長マニピュレータに対してもその有用性を確認することができた。

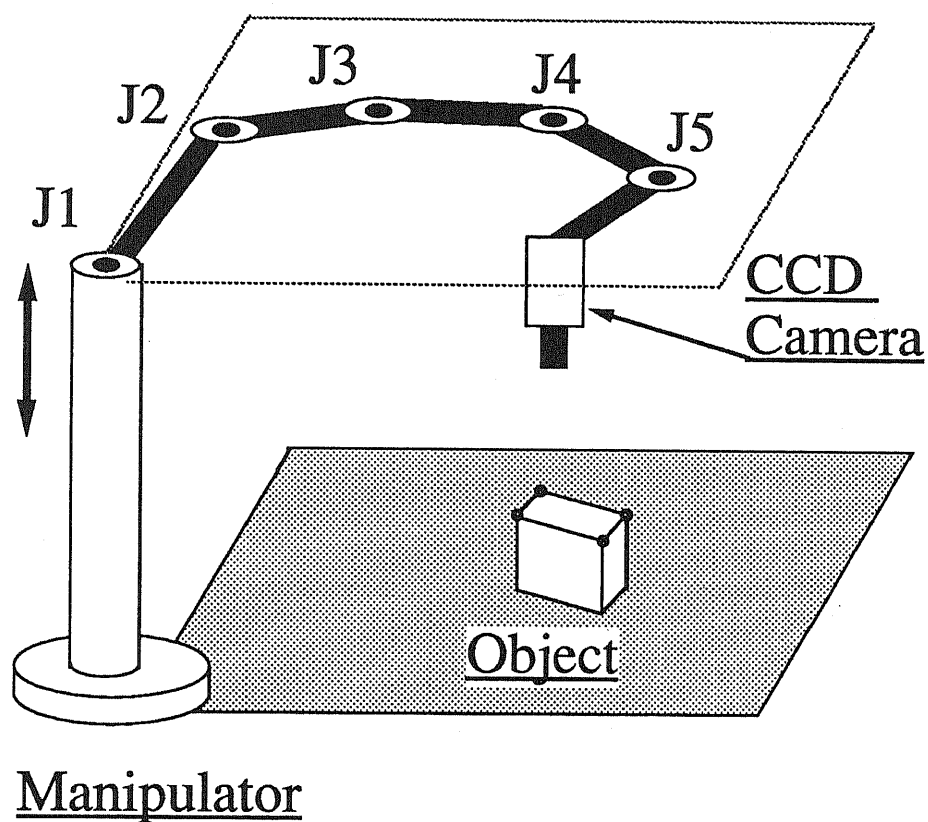


Figure A.42 Simulation model

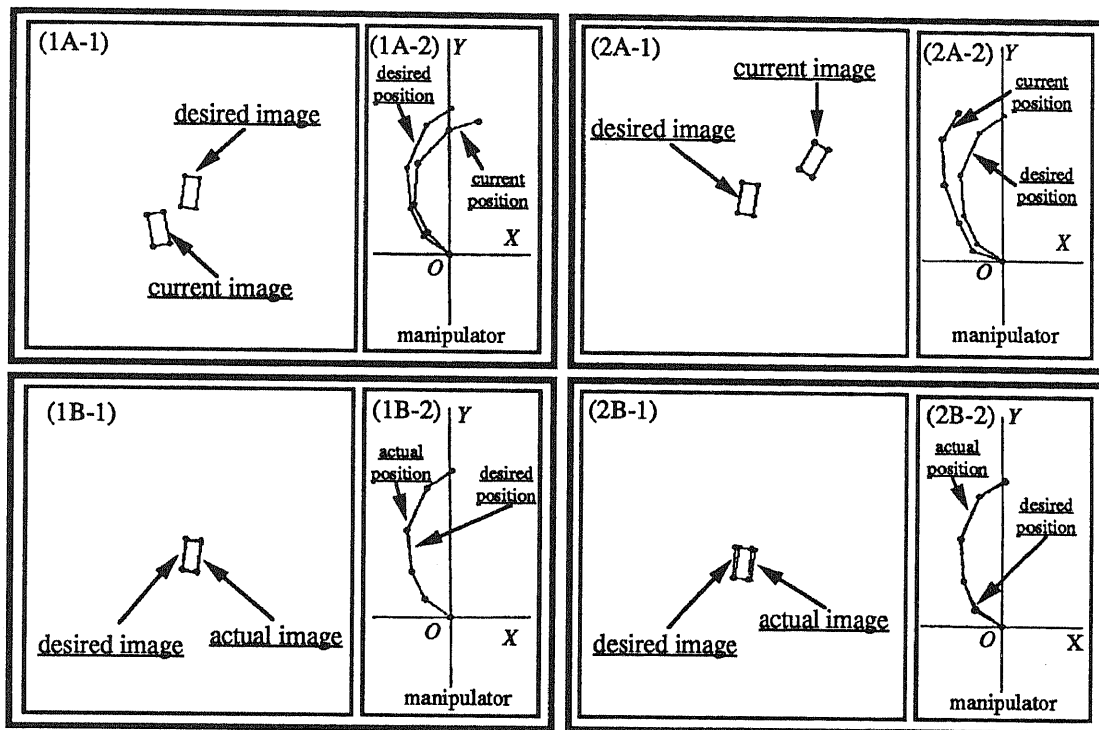


Figure A.43 Simulation results (for learned pattern)

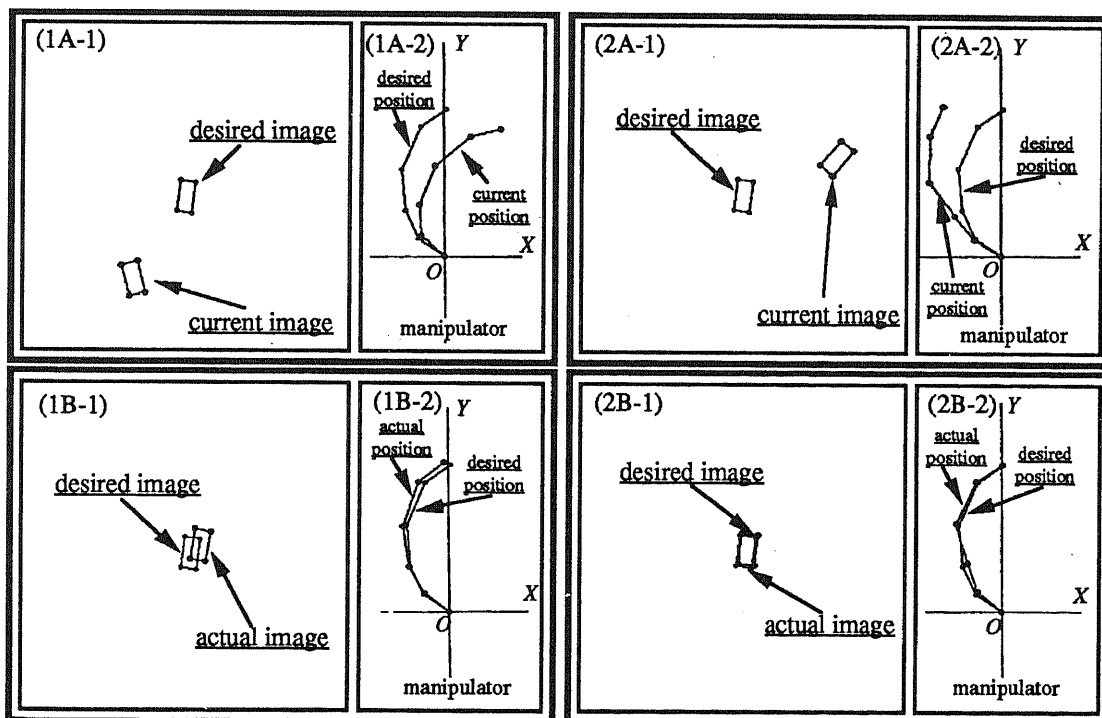


Figure A.44 Simulation results (for not-learning pattern)

## A.8 まとめ

マニピュレータの先端にカメラを取り付け、視覚情報のみを用いて目標位置、目標姿勢に先端を移動させる方法について述べた。ここでは、対象物→カメラ座標系→マニピュレータ先端座標系→関節角座標系というような複雑な座標変換をすることなく、直接視覚情報から各関節角への制御入力情報を得る方法を提案した。本手法によって、視覚情報と関節角度情報との関係を複雑な非線形方程式を解かずに、ニューラルネットワークによる簡単なマッピングによって実現することができる。実際に、6自由度マニピュレータを対象としてシミュレーションを行い、また5自由度マニピュレータを用いて実験を行い、提案した手法が有効であることを示した。

本手法は、複雑な幾何学的計算がニューラルネットワークの簡単なマッピングにより実現できるため、実時間処理に向いていると考えられる。また、視覚による大体の位置決めという観点からは本手法が有効であるといえる。さらに、マニピュレータの持つコンプライアンスと組み合わせて知的な作業を行わせることが期待できる。



## 参考文献

- [麻生 88] 麻生, "ニューラルネットワークの情報処理," 産業図書 (1988).
- [甘利 72] 甘利, "神経回路網の数理," 産業図書 (1972).
- [Albus75] J.S.Albus, "A New Approach to Manipulator Control : The Cerebellar Model Articulation Controller," Trans. of the ASME, pp.220-227 (1975).
- [Anderson85] R.L.Anderson, "Real Time Intelligent Visual Control of a Robot," IEEE Int. Workshop on Intelligent Control, pp.89-94 (1985).
- [Feddema89] J.T.Feddema, "Automatic Selection of Image Features for Visual Servoing of A Robot Manipulator," Proc. of IEEE Int. Conf. on Robotics and Automation, pp.832-837 (1989).
- [Guez88] A.Guez, Z.Ahmad, "Solution to the Inverse Kinematics Problem in Robotics by Neural Networks," Proc. of IEEE Int. Conf. on Neural Networks, pp.- (1989).
- [橋本 88] 橋本, 野田, 原島, "自己組織化システムの一構成法 (ニューラルネットのロボットアームへの適用)," 第 11 回 D S T シンポジウム, pp.211-216 (1988).
- [橋本 89a] 橋本, 久保田, 佐藤, 原島, "視覚情報によるマニピュレータの位置・姿勢制御 (ニューラルネットワークの利用)," 日本ロボット学会誌, Vol.8, No.4, pp.390-396 (1990).
- [Hashimoto89b] H.Hashimoto, T.Kubota, W.C.Lo, F.Harashima, "A Control Scheme of Visual Servo Control of Robotic Manipulators Using Artificial Neural Network," Proc. of IEEE Int. Conf. on Control and Applications, TA-3-6 (1989).
- [Hashimoto89c] H.Hashimoto, T.Kubota, M.Sato, F.Harashima, "Visual Servo Control of Robotic Manipulators Based on Artificial Neural Network," Proc. of IEEE Int. Conf. on Industrial Electronics, pp.770-774 (1989).
- [堀口 85] 堀口, "単眼視による既知平面図形の距離/姿勢の一測定法," 第 3 回日本ロボット学会学術講演会予稿集, pp.295-296 (1985).
- [Hung85] Y.Hung, P.Yeh, D.Harwood, "Passive Ranging to Known Planar Point Sets", Proc. of IEEE Int. Conf. on Robotics and Automation, pp.80-84 (1985).
- [川端 90] 川端, "k 近傍内挿学習による音韻認識," 日本音響学会講演論文集, pp.161-162 (1990).

- [Kano90] H.Kano, K.Takayama, "Learning Control of Robotic Manipulators Based on Neurological Model CMAC," Proc. of IFAC, pp.268-273 (1990).
- [Kawato87] M.Kawato, K.Furukawa, R.Suzuki, "A Hierarchical Neural Network Model for Control and Learning of Voluntary Movement," Biological Cybernetics 57, pp.169-185 (1989).
- [Kuperstein89] M.Kuperstein, "Neural Model of Adaptive Hand-Eye Coordination for Single Postures," Science, Vol.239, pp.1308-1311 (1989).
- [Kroese90] B.J.A.Kroese, M.J.Korst, F.C.A.Groen, "Learning Strategies for a Vision Based Neural Controller for a Robot Arm," Proc. of the IEEE Int. Workshop on Intelligent Motion Control, Vol.1, pp.198-207 (1990).
- [Martinetz90] T.M.Martinetz, H.J.Ritter, K.J.Schulten, "Three-Dimensional Neural Net for Learning Visuomotor Coordination of a Robot Arm," IEEE Trans. on Neural Networks, Vol.1, No.1, pp.131-136 (1990).
- [Miller87] W.T.Miller, "Sensor Based Control of Robotic Manipulators Using General Learning Algorithm," IEEE Journal of Robotics and Automation, Vol.RA3, No.2, pp.157-165 (1987).
- [Miller89] W.T.Miller, "Real-Time Application of Neural Networks for Sensor-Based Control of Robots with Vision," IEEE Trans. on Systems, Man, and Cybernetics, Vol.19, No.4 (1989).
- [Pao87] Y.H.Pao, D.J.Sobajic, "Connectionist-net Technology for Intelligent Robotics Control", Int. Workshop on Robotics: Trends, Technology and Applications, pp.9-16 (1987).
- [Rumelhart86] D.E.Rumelhart, J.L.McClelland, the PDP Research Group, "Parallel Distributed Processing", Bradford Books, Vol.1 & 2, MIT Press (1986).
- [Sanderson85] A.C.Sanderson, "Vision and Multisensor Feedback Controllers," IEEE Int. Workshop on Intelligent Control, pp.80-81 (1985).
- [佐藤 87] 佐藤, 長谷川, "視覚フィードバックによるロボットの制御," 計測と制御, Vol.26, No.2, pp.24-29 (1987).
- [田中 88] 田中, 清水, 土屋, "神経回路網によるマニピュレータの運動制御," 第3 2 回システムと制御講演会, pp.183-184 (1988).

- [Tsai84] R.Tsai, T.Huang, "Uniqueness and Estimation of Three Dimensional Motion Parameters of Rigid Objects with Curved Surfaces," IEEE Trans. on PAMI-6 (1984).
- [Weiss85] L.E.Weiss, A.C.Sanderson, "Dynamic Sensor-Based Control of Robots with Visual Feedback," Proc. of IEEE Int. Workshop on Intelligent Control, pp.102-109 (1985).

# Acknowledgement

謝 辭

本論文は、筆者が1986年春から1991年冬にかけて、東京大学大学院工学系研究科電気工学専攻課程に在学中に行った研究をまとめたものである。その間、指導教官原島文雄教授（東京大学生産技術研究所3部）には、常に高所から研究全体を把握され、貴重な御助言と暖かい御指導・励ましを賜りました。心からお礼を申し上げます。

また、研究に行き詰まったときに幾度となく相談にのっていただき、数々の的確な御意見、熱心な御指導をいただきました橋本秀紀助教授（東京大学生産技術研究所3部）に心から感謝いたします。

さらに、有益な御助言をいただいた東京大学生産技術研究所3部近藤正示助手、実験装置に関して御助言をいただいた芝浦工業大学の春日智恵講師に深く感謝致します。

本研究の実験の一部および本論文の完成に度々協力して下さった研究生工藤正明氏、実験の一部を手伝っていただいた当時受託研究員佐藤基夫氏、当時大学院学生の東辰輔氏、森田敏生氏、大学院学生の勝野聡氏、小國哲氏、金子智朗氏、また研究室の計算機環境を整えてくれた大学院学生のマーチンブス氏、さらに大学生生活を送るにあたり数々の相談にのってくれた大学院学生の古関隆章氏に感謝致します。

日頃の研究生生活におきまして公私にわたりお世話になりました原島・橋本研究室の皆様、海外からの研究者の方々に感謝の意を表してここに名前を列記します。本当にありがとうございました。最後に、研究生生活を全面的に協力し、本研究を大学にて自由に行う機会を与えて下さった両親に心から感謝致します。

元助手 稲葉博氏 技官 長谷川仁則氏 杉本賢司氏

大学院学生（当時含む）

許建新氏 沈貴宝氏 山本肇氏 野田浩氏 畔柳洋氏 姜聖允氏 勞偉籌氏  
梁承学氏 白文鴻氏 鈴木裕之氏 出口欣高氏

研究生（当時含む）

宮田勝氏 加藤洋治氏 金澤博之氏 遠藤淳一氏  
柳在補氏 陳衛平氏 Mr.B.Ricard

受託研究員（当時含む）

新井雄一氏 西山豊氏 柳沢重一氏 中山朝法氏 八軒英太郎氏  
大野雄幸氏 出水祐三氏 飯塚昇氏 井上修司氏 滝沢聡毅氏

客員研究員（当時）

Prof. A.Sabanovich Prof. V.I.Utkin Prof. O.Kaynak 沈廉教授

秘書（当時含む）

高岸寿子女史 吉田みどり女史 平木邦子女史  
渥美容子女史 川本泰子女史 松原明子女史

なお、本研究の一部は文部省科学研究費補助金奨励研究(A)01790388による。

signature 久保田 孝

# Publication

## 発表論文リスト

## A. 論文誌

- [1] 久保田, 橋本, 原島, "ローカルプランニングによる移動ロボットの経路探索," 日本ロボット学会誌, Vol.7, No.4, pp.267-274 (1989).
- [2] 久保田, 橋本, 原島, "視覚センサとレンジセンサとの協調による移動ロボットのナビゲーション," 日本ロボット学会誌, Vol.7, No.4, pp.275-283 (1989).
- [3] 橋本, 久保田, 佐藤, 原島, "視覚情報によるマニピュレータの位置・姿勢制御 (ニューラルネットワークの利用)," 日本ロボット学会誌, Vol.8, No.4, pp.390-396 (1990).
- [4] T.Kubota, H.Hashimoto, F.Harashima, "Path Searching for Mobile Robot by Local Planning," ADVANCED ROBOTICS Int. Journal of the Robotics Society of Japan に掲載決定.
- [5] T.Kubota, H.Hashimoto, F.Harashima, "Control of Mobile Robot Based on Cooperation of Sensors," IEEE Trans. on IE に投稿中.
- [6] H.Hashimoto, T.Kubota, M.Sato, F.Harashima, "Visual Control of Robotic Manipulator Based on Neural Networks," IEEE Trans. on IE に投稿中.

## B. 国際会議

- [1] T.Kubota, H.Hashimoto, F.Harashima, "Control of Mobile Robot Based on Cooperation of Vision and Range Sensors," Int. Workshop on Robotics : Trends, Technology and Applications, Madrid SPAIN (1987).
- [2] H.Hashimoto, T.Kubota, F.Harashima, "Path Planning of Mobile Robot Based on Visual Information," USA-JAPAN Symposium on Flexible Automation, Minneapolis U.S.A. (1988).
- [3] T.Kubota, H.Hashimoto, F.Harashima, "Path Searching of Mobile Robot Based on Cooperation of Sensors," IEEE Int. Workshop on Intelligent Robots and Systems (Toward the Next Generation Robot and System), Tokyo (1988).
- [4] H.Hashimoto, T.Kubota, W.C.Lo, F.Harashima, "A Control Scheme of Visual Servo Control of Robotic Manipulators Using Artificial Neural Network," IEEE Int. Conf. on Control and Applications, Jerusalem ISRAEL (1989).
- [5] H.Hashimoto, T.Kubota, M.Sato, F.Harashima, "Visual Servo Control of Robotic Manipulators Based on Artificial Neural Network," IEEE Int. Conf. on Industrial Electronics, Philadelphia U.S.A. (1989).
- [6] T.Kubota, H.Hashimoto, "A Strategy for Collision Avoidance among Moving Obstacles for a Mobile Robot," 11th World Congress of International Federation of Automatic Control, Tallin, Estonia USSR (1990).
- [7] F.Harashima, H.Hashimoto, T.Kubota, "Sensor Based Robot Control Systems," IEEE Int. Workshop on Intelligent Motion Control, Istanbul TURKEY (1990).
- [8] H.Hashimoto, T.Kubota, F.Harashima, "Visual Control of a Robotic Manipulator Using Neural Networks," the 29th IEEE Conf. on Decision and Control, Hawaii U.S.A. (1990).
- [9] H.Hashimoto, T.Kubota, M.Kudou, F.Harashima, "Self-Organizing Visual Servo System Based on Neural Networks," 1991 American Control Conference (1991) (to be presented).



## C. 全国大会

- [1] 久保田, 橋本, 原島, "移動ロボットにおけるセンサ情報処理," 第26回計測自動制御学会学術講演会, No.JS49-2 (1987).
- [2] 久保田, 橋本, 原島, "階層的プランニングによる経路探索(実験的検証)," 第5回日本ロボット学会学術講演会, No.2405 (1987).
- [3] 久保田, 橋本, 原島, "単眼視による移動ロボットの複数障害物回避(回避ポイント生成のための画像処理アルゴリズム)," 第27回計測自動制御学会学術講演会, No.JS37-3 (1988).
- [4] 久保田, 橋本, 原島, "センサ情報融合による移動ロボットのナビゲーション," 第6回日本ロボット学会学術講演会, No.2510 (1988).
- [5] 久保田, 橋本, 原島, "視覚センサを用いた移動ロボットのローカルパスプランニング," 第28回計測自動制御学会学術講演会, No.JS3-3 (1989).
- [6] 橋本, 久保田, 佐藤, 原島, "マニピュレータのビジュアルサーボコントロール(ニューラルネットワークの利用)," 第28回計測自動制御学会学術講演会, No.JS64-1 (1989).
- [7] 久保田, 橋本, 原島, "視覚情報を用いた移動ロボットの障害物回避(移動障害物に対する回避行動の形成)," 第7回日本ロボット学会学術講演会, No.1103 (1989).
- [8] 橋本, 久保田, 佐藤, 原島, "視覚情報によるマニピュレータの位置・姿勢制御(神経回路網の利用)," 第7回日本ロボット学会学術講演会, No.3507 (1989).
- [9] 久保田, 橋本, 原島, "センサの協調に基づく移動ロボットのナビゲーション," 第2回ロボットセンサシンポジウム, No.1303 (1990).
- [10] 久保田, 橋本, 原島, "移動ロボットの障害物回避のための状況判断・行動決定に関する考察," 第29回計測自動制御学会学術講演会, JS41-2 (1990).
- [11] 橋本, 久保田, 原島, "Neural Networkを用いたマニピュレータのビジュアル制御," 第29回計測自動制御学会学術講演会, JS37-4 (1990).

- [12] 久保田, 橋本, 原島, "移動障害物に対する回避行動形成の一手法," 第8回日本ロボット学会学術講演会, No.1501 (1990).
- [13] 橋本, 久保田, 原島, 工藤, "ニューラルネットを用いたマニピュレータの位置・姿勢制御(実験的検討)," 第8回日本ロボット学会学術講演会, No.1117 (1990).
- [14] 橋本, 久保田, "センサ情報に基づく移動ロボットの障害物回避," 第8回センシングフォーラム (1991) (発表予定).

## D. 研究会

- [1] 久保田, 橋本, 原島, "ロボットシステムにおける知的制御," 生産技術研究奨励会第46研究委員会(知的運動制御とその高度化)第2回研究会 (1987).
- [2] 久保田, 橋本, 原島, "センサの協調による移動ロボットのパスプランニング," 第4回知能ロボット・オートメーション研究分科会 (1989).
- [3] 久保田, 橋本, 原島, "ニューラルネットによるロボットのビジュアル制御," 画像応用技術専門委員会 1989年度第4回研究会 (1990).

## E. その他

- [1] 原島, 久保田, 橋本, "ロボットシステムにおける知的制御," 東京大学生産技術研究所電気談話会報告, Vol.37, No.1 (1987).
- [2] 久保田, 橋本, 原島, "センサの協調による移動ロボットのナビゲーション," 東京大学生産技術研究所電気談話会報告, Vol.39, No.3 (1989).
- [3] 久保田, 佐藤, 橋本, 原島, "ニューラルネットを用いたロボットのビジュアル制御," 東京大学生産技術研究所電気談話会報告, Vol.40, No.4 (1990).
- [4] 橋本, 久保田, "移動ロボットの障害物回避," 機械の研究, 第43号, 第1号, pp.151-pp.157 (1991).