

## 視覚情報を用いたロボットマニピュレータ 作業制御システムの試作

渡辺 栄治\*、清水 光\*

### Experimental Task Control System of Robot Manipulator Using Visual Information

Eiji WATANABE and Hikaru SHIMIZU

#### ABSTRACT

As a trial of the realization of the intelligent robot control system, we construct a task control system of robot manipulator using visual information and show the effectiveness of the system by blocks rearrangement experiments.

The task control system consists of following three sub systems.

- (1) Visual information processing system : The blocks in the initial state are recognized based on visual information acquired from CCD(Charge Coupled Device) camera.
- (2) Task planning system : The task procedures of blocks rearrangement are generated from the task procedures data base described by the robot macro language.
- (3) Task executing system : The task procedures are translated into robot manipulator commands, and the blocks rearrangement task can be achieved by robot manipulator.

Finally, we show the effectiveness of the proposed task control system of robot manipulator from the experimental results of blocks rearrangement task.

Key words : Task control of robot manipulator, Visual information processing, Task planning, Task executing, Blocks rearrangement experiment

#### 1. 緒言

1960年代以降、プレイバック方式の産業用ロボットが製造業を中心として産業界のあらゆる分野で使用され、工場の自動化に貢献してきた。プレイバック方式のロボットは、あらかじめ人間に教えられた作業内容・手順を記憶装置等に記憶しておき、その作業を正確に繰り返すことを目的として開発された。しかし、ロボットの環境が教示された内容と異なる場合、上述の方式では正確な動作が実行不可能となる。そこで、視覚、聴覚、触覚セン

サー等により環境の変化を認識し、センサー情報から、自らの行動を決定できる能力を持つロボットが必要となる。現在、環境の変化に適応可能な知能ロボットの制御方式に関する基礎・応用研究が活発に行われている<sup>(1)-(3)</sup>。

本論文では、パーソナルコンピュータによる知能ロボット制御システムの実現の試みとして、視覚情報を利用したロボットマニピュレータ作業制御システムを構築し、積木の並べ替え問題に適用し、その有効性を示す。まず、2節において、4個の積木を用いた並び替え問題につい

て述べる。次に、3節では、視覚情報を利用したロボットマニピュレータ作業制御システムを提案する。4節では、各積木の位置、方向、種類及び、積木間の相対関係等の認識手法について述べる。5節では、積木を初期状態から指定された目標状態に並べ替えるためのタスクを生成する手法について述べる。6節では、ロボットマニピュレータに与えるコマンド群を生成する手法について述べる。最後に、7節では、ここで提案する作業制御システムを用いて、4個の積木を用いた並び替え作業実験を実施し、本システムの有効性を示す。

## 2. 問題記述

本節では、本論文で対象とする積木並び替え問題について述べる。ここでは、図1に示すような4個の柱状の積木を用いる。まず、平面上に任意の順序で積木を並べ、この状態を初期状態とする。次に、外部から積木の順序が指定され、この状態を目標状態とする。問題は、初期状態の積木を認識し、目標状態に並べ替える作業手順を生成することである。なお、初期状態において、積木同士の接触はなく、また、積木は同一平面上に設置されるものと仮定する。図2に積木並び替え問題の具体例を示す。ここで、初期状態における積木の設置エリアを、左から $P_1$ ,  $P_2$ ,  $P_3$ ,  $P_4$ と定めている。初期状態として、左からA, B, C, Dの順序(図2(a))で並んでいる積木を、C, D, B, Aの順序(図2(b))に並び替える例を表している。また、初期状態から目標状態に到達する為の作業手順を図2(c)~(g)に示す。実際には、初期状態の積木の並びは未知であり、また、初期状態と目標状態の組合せ総数 $24^2$ 通りから、適当な作業手順を決定しなければならない。従って、本問題を解決する為には、以下の様な機能を有するシステムを開発する必要がある。

- ・初期状態における各積木の位置、方向、種類、状態及び積木間の相対関係等の認識。
- ・初期状態から目標状態に到達するための作業手順の生成。
- ・作業手順からロボットコマンドへの変換。

以下では、上記の機能を実現する為のロボット作業制御システムについて述べる。

## 3. 視覚情報を利用したロボット作業制御システム

### 3.1 ロボット作業制御システムのハードウェア構成

図3に本システムのハードウェア構成を示す。ホストコンピュータとして、32bitパーソナルコンピュータ(日本電気製PC9801-RA51)を使用し、ロボットマニピュレータは、三菱電機(株)製RM101を用いる。視覚情報処理部は、CCDカメラ(SONY製HVM-52C)2台及び、画像入力ボード(ソアテック製VIDEO98、512\*512画素、16階調)2枚から構成されている。CCDカメラは、各々、作業エリアに対して垂直・水平方向に固定されている。

### 3.2 ロボット作業制御システムのソフトウェア構成

以下の事項を満足する様にロボット作業制御システムを構成している。本システムでは、視覚情報処理部、作業

- ・作業手順の教示が容易。
- ・システム構成が作業内容に依存しない。
- ・作業対象物の変更に伴うシステムの修正が小さい。

計画部、作業実行部の3つのサブシステムから構成される。視覚情報処理部では、2台のCCDカメラから入力される視覚情報を基に、画像処理手法を用いて積木の対角線長や辺長を算出し、積木の特徴量データベースを参照して、各積木の位置、方向、種類、状態及び積木間の

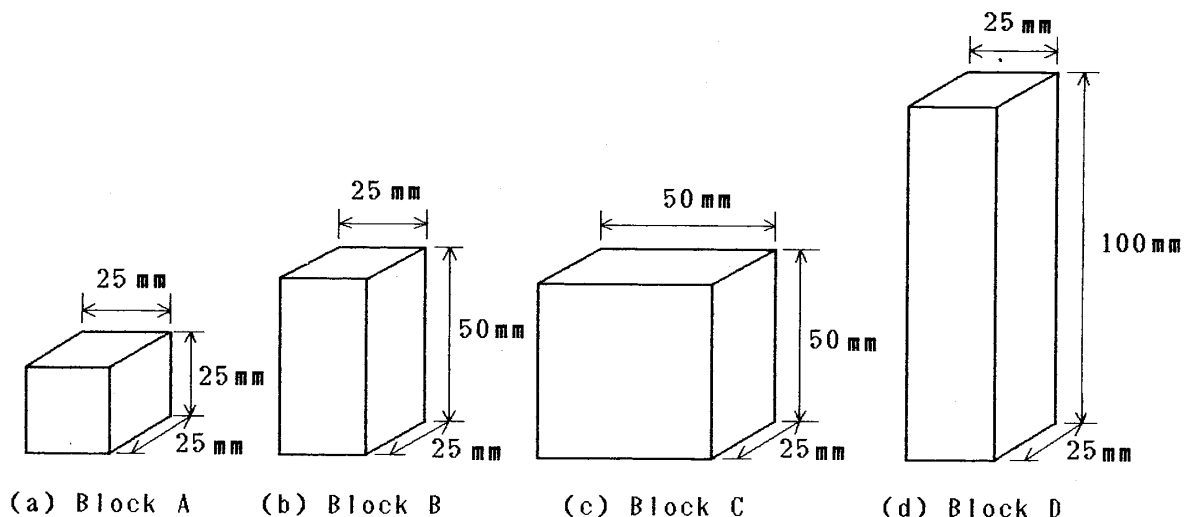


Fig.1 Size of blocks

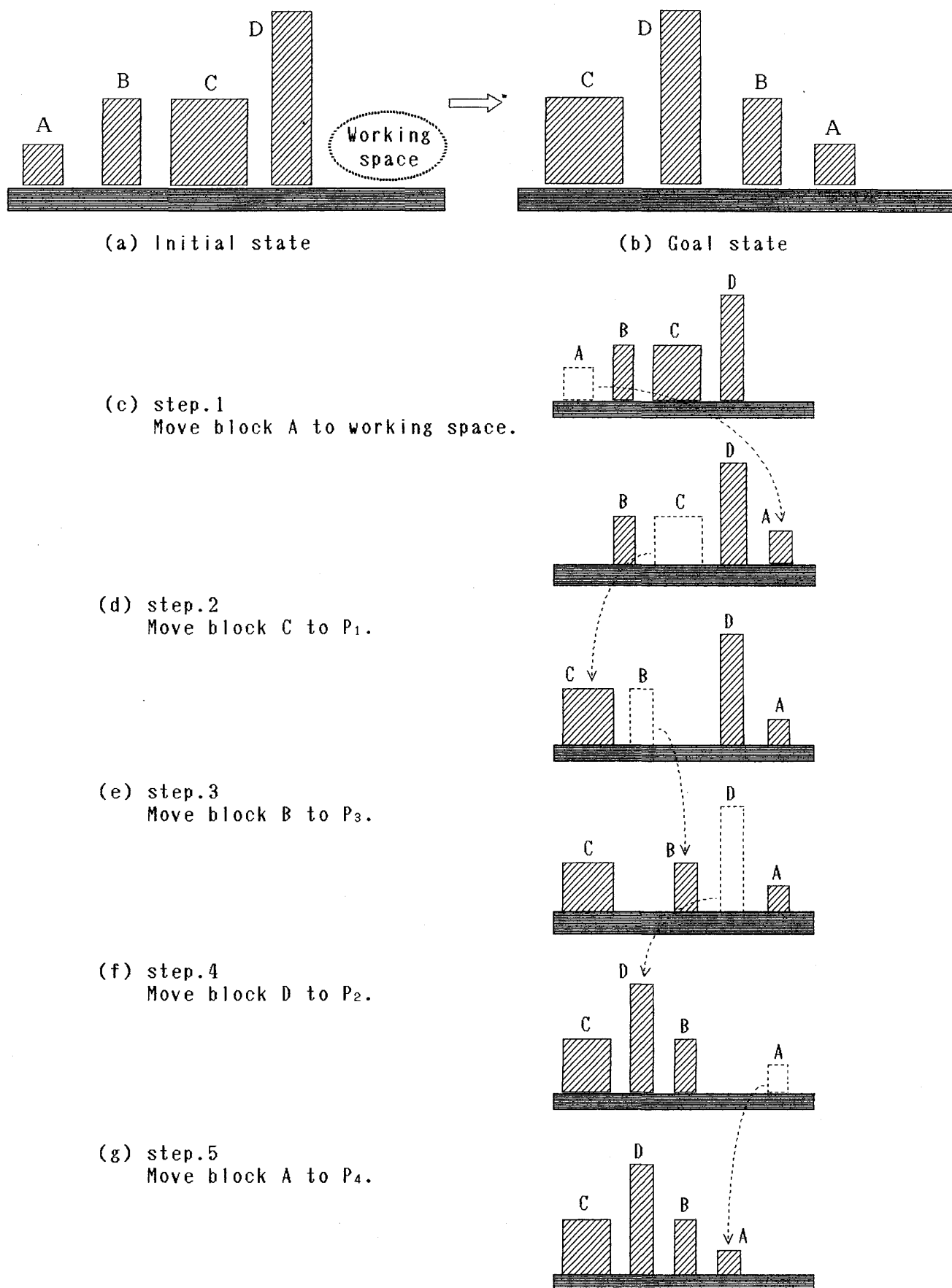


Fig.2 An example of rearrangement task

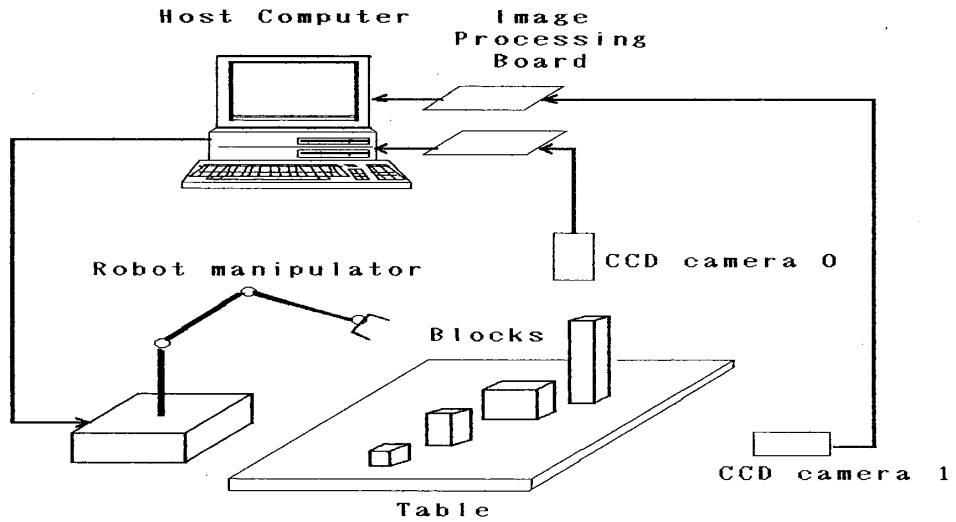


Fig.3 Configuration for task control system of robot manipulator using visual information

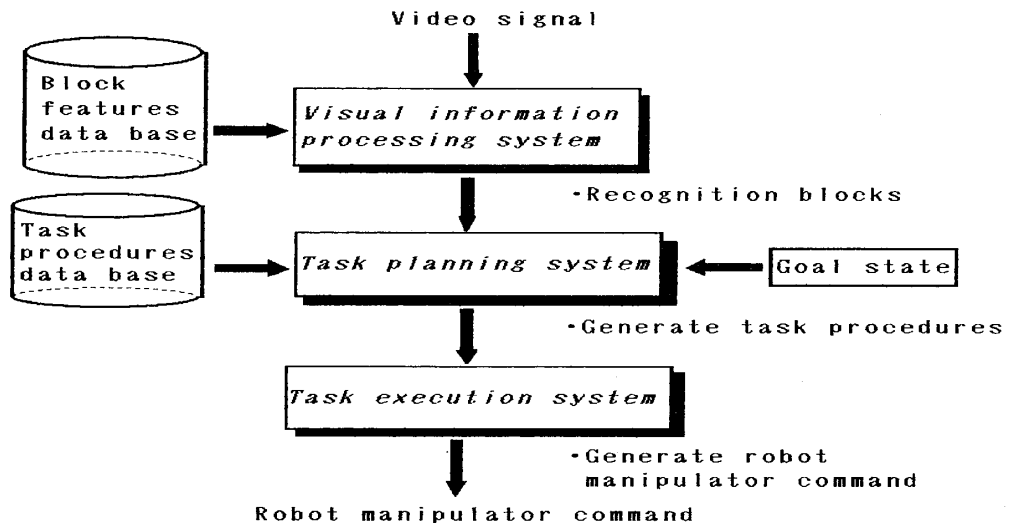


Fig.4 Blockdiagram of robot task control system

相対関係等を認識する。次に、作業計画部では、積木の認識結果から、積木の初期状態と目標状態との差を検出し、積木を並べ替えるための作業手順を作業手順データベースを用いて生成する。最後に、作業実行部では、作業計画部で生成された作業手順をサブタスクに分解し、ロボットマニピュレータに与えるコマンド群を生成し、

積木の並び替え作業を実行する。システム全体の処理フローを図4に示す。以下では、サブシステム各部について説明する。

#### 4. 視覚情報処理部

本節では、2台のCCDカメラから入力される視覚情

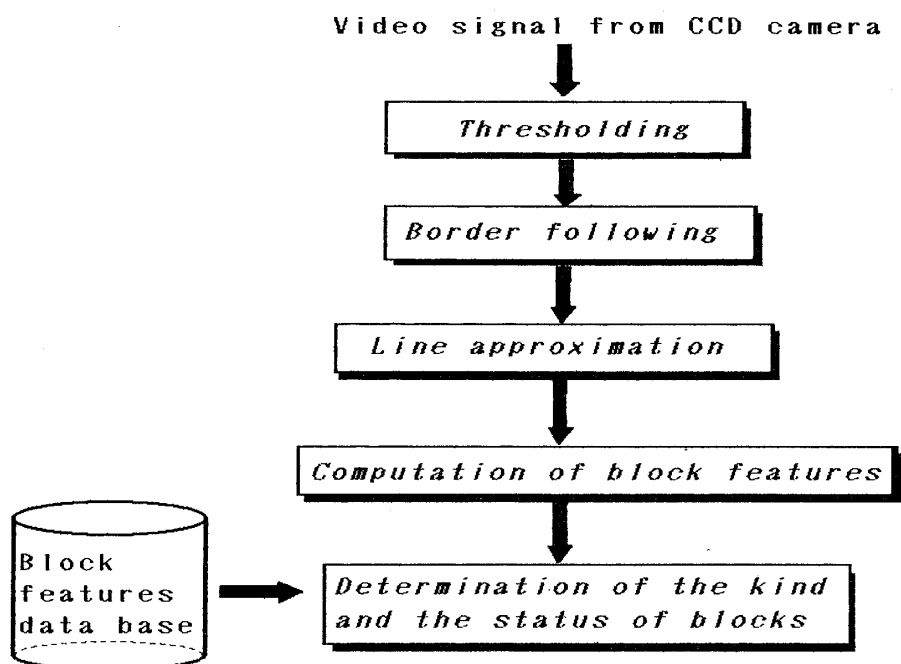


Fig.5 Flow of visual information processing system

報を基に、各積木の位置、方向、種類、状態及び積木間の相対関係等を認識するためのアルゴリズムについて述べる。また、本アルゴリズムの処理フローを図5に示す。

#### 4.1 積木の位置、種類、方向及び、状態の認識

まず、画像データ量の圧縮の為、垂直・水平方向の2台のCCDカメラから得られる16階調の画像データに対して2値化処理を行う。次に、2値化画像データに対して、輪郭追跡を行い、図形の抽出を行う。次に、図6に示す直線近似法を用いて、各輪郭点から直線までの最大垂直距離  $\delta_{max}$  が許容範囲ならば、冗長な輪郭点を除去し、線図形の整形化を行う<sup>(4)</sup>。さらに、整形化された線図形データより、図7で定義される重心座標、方向、各辺の長さ及び、重心座標から頂点までの距離を以下の様に算出する。

##### 重心 $(x_G, y_G)$

積木の重心は、図形の輪郭点系列  $\{P_i(x_i, y_i), i=1, \dots, 4\}$  より、次式で与えられる。

$$x_G = \frac{\sum_{i=1}^4 x_i}{4}, \quad y_G = \frac{\sum_{i=1}^4 y_i}{4} \quad (4.1)$$

##### 積木の方向

まず、各頂点の角度を次式で定義する。

$$\theta_i = \tan^{-1}((y_{i+1} - y_i) / (x_{i+1} - x_i)) \quad (4.2)$$

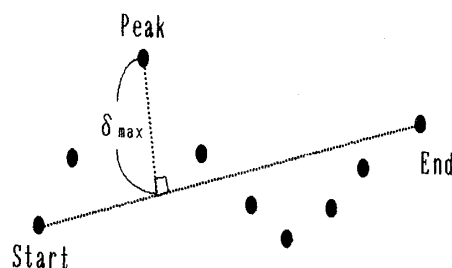


Fig.6 Line approximation method

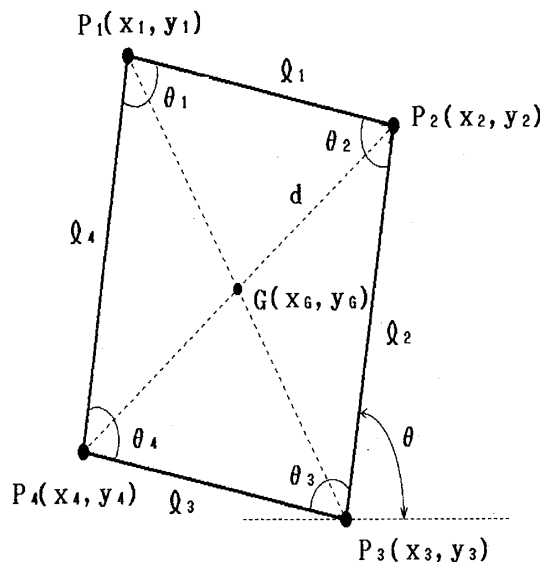


Fig.7 Features of block

Table 1 Blocks features data base

Blocks recogniton rule	
Rule for determination of the status ("stand" or "lie")	
Block A	If $(7 \leq D_0 \leq 10)$ and $(7 \leq D_1 \leq 10)$ "lie"
Block B	If $(7 \leq D_0 \leq 14)$ and $(6 \leq D_1 \leq 14)$ or $(0 \leq  D_0 - D_1  \leq 2)$ and $(\theta \leq 45^\circ)$ If $u \leq 0.65$ then "lie" else "stand"
Block C	If $(12 \leq D_0 \leq 17)$ and $(12 \leq D_1 \leq 18)$ or $(0 \leq  D_0 - D_1  \leq 2)$ and $(\theta > 45^\circ)$ If $u \leq 0.65$ then "stand" else "lie"
Block D	If $(7 \leq D_0 \leq 30)$ and $(7 \leq D_1 \leq 30)$ If $u \leq 0.65$ then "lie" else "stand"

where  $D_i$  shows the distance between the centroid and the vertex of image data from the  $i$ -th camera

4頂点のうち、最も $90^\circ$ に近い頂点をはさむ2辺を抽出し、2辺のうち長い辺の水平方向に対する角度を積木の方向 $\theta$ とする。

#### 最も $90^\circ$ に近い頂点をはさむ2辺の比

最も $90^\circ$ に近い頂点をはさむ2辺の比 $u$ を次式で定義する。

$$u = l_{\min} / l_{\max} \quad (4.3)$$

ここで、 $l_{\min}$ 、 $l_{\max}$ は、最も $90^\circ$ に近い頂点をはさむ2辺の長さを表す。但し、 $l_{\min} \leq l_{\max}$ とする。

#### 重心座標から頂点までの距離

重心座標から頂点までの距離は次式で与えられる。

$$D = \sqrt{(x_G - x_1)^2 + (y_G - y_1)^2} \quad (4.4)$$

#### 積木の相対関係の決定

(4.4)式で算出された各図形の重心座標 $(x_G, y_G)$ の中で、 $x$ 座標の小さな順に $P_1, P_2, P_3, P_4$ と定める。

(4.1)~(4.4)式で定義された図形特徴量と表1に示す積木の特徴量データベースとの内容を比較し、積木の種類を決定する。ただし、図1で示されるように、垂直方向のCCDカメラの視覚情報だけでは、積木の方向や状態により、他の積木と誤認識する場合がある。そこで、水平方向から算出された特徴量を参照することにより、積木を識別できないことを防いでいる。また、各図形の重心位置から、積木の並びを決定する。さらに、ロボットに適当な高さで積木を把握させるために垂直方向のCCDカメラから得られた図形の特徴量から積木の状態(立った状態、座った状態)を認識する。

#### 4.2 カメラ座標系からロボット座標系への変換

実際に、ロボットにより、積木を把握させるためには、前節で求められた積木の位置をカメラ座標系から、ロボットの作業環境内で定義されるロボット座標系に変換する必要がある。図8に、カメラ座標系とロボット座標系の関係を示す。ここで、積木は平面上に位置するために、<sup>(4)</sup>次式で与えられるアフィン変換により、ロボット座標系

における積木の位置を求める。

$$r_i = R r_{Gi} + T, \quad i=1,2,3 \quad (4.5)$$

但し、回転行列 $R$ 、移動ベクトル $T$ 、カメラ座標系、ロボット座標系における重心ベクトル $r_{Gi}$ 、 $r_i$ は以下の様に与えられる。

$$R = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, T = \begin{pmatrix} e \\ f \end{pmatrix}, r_{Gi} = \begin{pmatrix} x_{Gi} \\ y_{Gi} \end{pmatrix}, r_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix}$$

ここで、 $\{(x_i, y_i), i=1,2,3\}$ はロボット座標系における既知なる3点であり、 $\{(x_{Gi}, y_{Gi}), i=1,2,3\}$ は3点をCCDカメラで入力した場合のカメラ座標系の対応点である。従って、回転行列 $R$ 、移動ベクトル $T$ を決定することにより、カメラ座標系からロボット座標系への変換が可能となる。ここで、(4.4)式の未知係数 $a, b, c, d, e$ 及び $f$ は、以下の連立方程式より求められる。

$$\begin{cases} x_1 = a x_{G1} + b y_{G1} + e \\ x_2 = a x_{G2} + b y_{G2} + e \\ x_3 = a x_{G3} + b y_{G3} + e \end{cases} \quad (4.6)$$

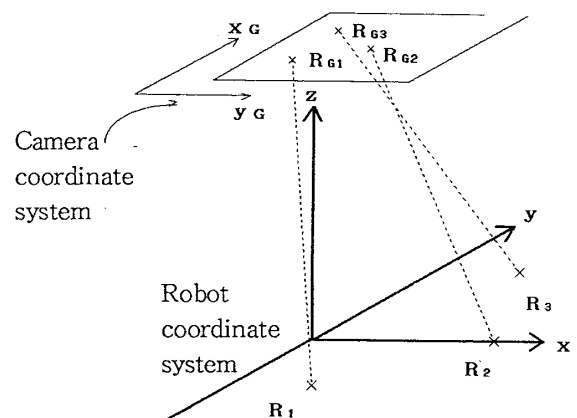


Fig.8 Camera and robot coordinate systems

$$\begin{cases} y_1 = c x_{G1} + d y_{G1} + f \\ y_2 = c x_{G2} + d y_{G2} + f \\ y_3 = c x_{G3} + d y_{G3} + f \end{cases} \quad (4.7)$$

本節で述べた画像処理手法を用いて図9の様に並べられた積木の認識結果を図10、表2に示す。図10(a)(b)は、各々、カメラ0、カメラ1の画像データに対する輪郭抽出

出の処理結果を示す。また、図10(c)(d)は、各々、カメラ0、カメラ1の画像データに対する区分的直線近似法による整形化の処理結果を示す。さらに、表2(a)(b)(c)に示すように積木の位置、方向、種類、状態が正しく認識されていることがわかる。

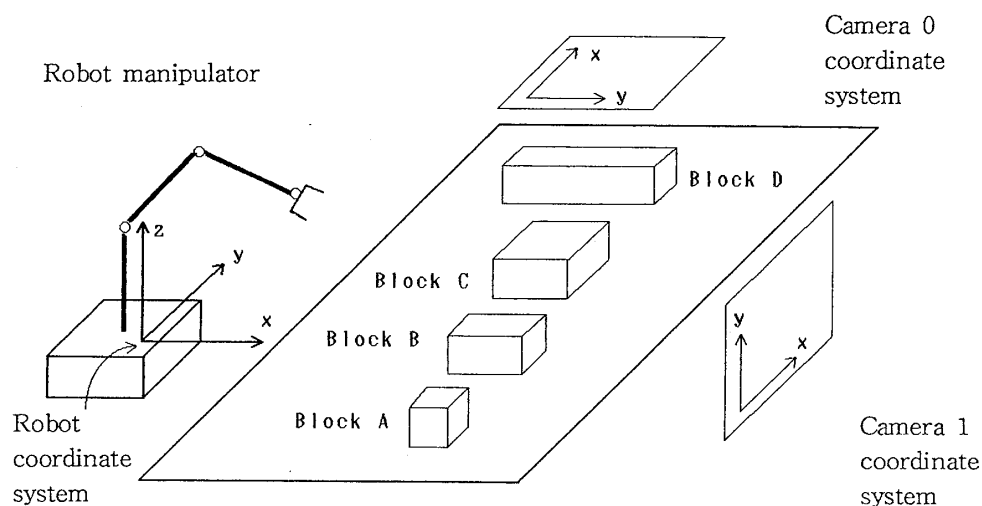
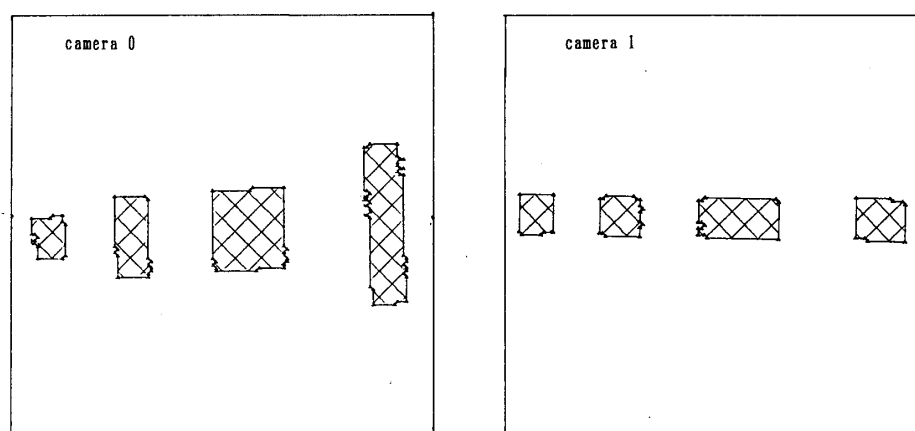
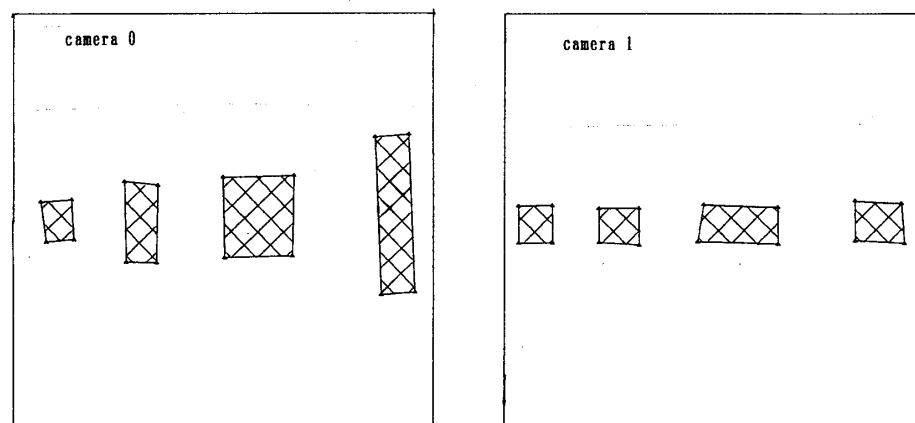


Fig.9 An example of recognition blocks



(a) Line-drawing by border following method



(b) Line-drawing by line approximation method

Fig.10 Processing result for the example shown in Fig.9

Table 2 Recognition results of blocks

(a) Translation vector and rotation matrix				
		Camera 0		
Translation vector T		(-595.6055, -580.2590)		
Rotation matrix R		0.0683	2.1062	
		2.3794	0.0313	

(b) Computation results from camera 0 and camera 1				
	camera 0			
	Image 1	Image 2	Image 3	Image 4
Image center	(198,414)	(225,414)	(263,414)	(308,410)
Distance D	8	14	17	26
Direction	85°	90°	88°	86°
	camera 1			
	Image 1	Image 2	Image 3	Image 4
Image center	(224,179)	(251,178)	(290,178)	(335,179)
Distance D	8	9	14	10
Direction $\theta$	90°	0°	2°	4°

(c) Recognition results				
	Image 1	Image 2	Image 3	Image 4
True center position	(310, -120, 85)	(310, -50, 85)	(310, 50, 85)	(310, 170, 85)
Estimated Center position	(308, -118, 85)	(312, -47, 85)	(311, 52, 85)	(315, 170, 85)
Position error	( 2, -2, 0)	(-2, -3, 0)	(-1, -2, 0)	(-5, 0, 0)
True direction	90°	90°	90°	90°
Estimated direction	85°	90°	88°	86°
Direction error	5°	0°	2°	4°
Kind of blocks	Block A	Block B	Block C	Block D
State	lie	lie	lie	lie

### 5. 作業計画部

視覚情報処理部において、各積木の位置、方向、種類、状態及び積木間の相対関係等が認識される。作業計画部では、認識された積木の初期状態と外部から与えられる目標状態との差を比較することにより、並び替え作業のための作業手順を作業手順データベースを用いて生成する。

まず、認識結果から初期状態と目標状態の積木の並びを比較し、移動すべき積木の個数Nを次式で決定する。

$$N = \sum_{i=1}^4 f [I_i - G_i] \quad (5.1)$$

但し、

$$f [x] = \begin{cases} 1 & x \neq 0 \\ 0 & x = 0 \end{cases} \quad (5.2)$$

(5.1)式で、 $I_i$ 、 $G_i$ は各々、初期状態、目標状態における積木の種類を表す。次に、移動する積木と移動しない積木及び、それらの場所を決定する。ここで、認識された積木の重心のx座標が小さい順に、位置番号を $P_i$ 、各々の積木の方向、状態（立った状態、座った状態）を

$S_i$ 、さらに、ワークエリアについてはWを記号として割り当てる。ここで、一旦、ワークエリアに退避した積木を再度、移動する作業が必要である為、積木の状態 $S_i$ を管理する必要がある。移動すべき積木の個数と積木の移動先の条件により、作業手順データベースから、適合する作業手順を抽出する。作業手順データベースは、表3の様な形式で記述されている。ここで、4個の積木を並べ替える組合せは24通り存在するが、移動すべき積木の個数と積木の移動先の条件により、12通りに集約される。表3に示すマクロ言語では、「MOVE  $P_i S_i P_j S_j$ 」、「EXCHANGE  $P_i S_i P_j S_j$ 」は、各々、「エリア $P_i$ にある状態 $S_i$ の積木をエリア $P_j$ に状態 $S_j$ で移動しなさい。」、「ワークエリアを利用して、エリア $P_i$ にある状態 $S_i$ の積木とエリア $P_j$ にある状態 $S_j$ の積木をワークエリアを用いて入れ替えなさい。」という作業を表している。ここで、移動すべき積木の個数、積木の移動先の条件及び、積木並び替え作業に関する記述をC言語により開発したマクロ言語で表すことにより、作業制御プログラムの作成負荷が大幅に軽減されている。



Table 3 Task procedures data base

Rule No.	N	Comparison condition	Task procedures				
			Action	Object	Status	Object	Status
1	2		EXCHANGE	P <sub>1</sub>	S <sub>1</sub>	P <sub>2</sub>	S <sub>2</sub>
2	3	If I <sub>1</sub> ≠ G <sub>2</sub> or I <sub>2</sub> ≠ G <sub>3</sub>	MOVE	P <sub>1</sub>	S <sub>1</sub>	W	S <sub>1</sub>
			MOVE	P <sub>2</sub>	S <sub>2</sub>	P <sub>1</sub>	S <sub>1</sub>
			MOVE	P <sub>3</sub>	S <sub>3</sub>	P <sub>2</sub>	S <sub>2</sub>
			MOVE	W	S <sub>1</sub>	P <sub>3</sub>	S <sub>3</sub>
3	3	If I <sub>2</sub> ≠ G <sub>1</sub> or I <sub>3</sub> ≠ G <sub>2</sub>	MOVE	P <sub>1</sub>	S <sub>1</sub>	W	S <sub>1</sub>
			MOVE	P <sub>3</sub>	S <sub>3</sub>	P <sub>1</sub>	S <sub>1</sub>
			MOVE	P <sub>2</sub>	S <sub>2</sub>	P <sub>3</sub>	S <sub>3</sub>
			MOVE	W	S <sub>1</sub>	P <sub>2</sub>	S <sub>2</sub>
4	4	If I <sub>1</sub> = G <sub>4</sub> and I <sub>2</sub> = G <sub>1</sub>	MOVE	P <sub>1</sub>	S <sub>1</sub>	W	S <sub>1</sub>
			MOVE	P <sub>2</sub>	S <sub>2</sub>	P <sub>1</sub>	S <sub>1</sub>
			MOVE	P <sub>3</sub>	S <sub>3</sub>	P <sub>2</sub>	S <sub>2</sub>
			MOVE	P <sub>4</sub>	S <sub>4</sub>	P <sub>3</sub>	S <sub>3</sub>
			MOVE	W	S <sub>1</sub>	P <sub>4</sub>	S <sub>4</sub>
5	4	If I <sub>1</sub> = G <sub>3</sub> and I <sub>2</sub> = G <sub>1</sub>	MOVE	P <sub>1</sub>	S <sub>1</sub>	W	S <sub>1</sub>
			MOVE	P <sub>2</sub>	S <sub>2</sub>	P <sub>1</sub>	S <sub>1</sub>
			MOVE	P <sub>4</sub>	S <sub>4</sub>	P <sub>2</sub>	S <sub>2</sub>
			MOVE	P <sub>3</sub>	S <sub>3</sub>	P <sub>4</sub>	S <sub>4</sub>
			MOVE	W	S <sub>1</sub>	P <sub>3</sub>	S <sub>3</sub>
6	4	If I <sub>1</sub> = G <sub>4</sub> and I <sub>2</sub> = G <sub>3</sub> and I <sub>3</sub> = G <sub>1</sub>	MOVE	P <sub>1</sub>	S <sub>1</sub>	W	S <sub>1</sub>
			MOVE	P <sub>3</sub>	S <sub>3</sub>	P <sub>1</sub>	S <sub>1</sub>
			MOVE	P <sub>2</sub>	S <sub>2</sub>	P <sub>3</sub>	S <sub>3</sub>
			MOVE	P <sub>4</sub>	S <sub>4</sub>	P <sub>2</sub>	S <sub>2</sub>
			MOVE	W	S <sub>1</sub>	P <sub>4</sub>	S <sub>4</sub>
7	4	If I <sub>1</sub> = G <sub>2</sub> and I <sub>2</sub> = G <sub>4</sub>	MOVE	P <sub>1</sub>	S <sub>1</sub>	W	S <sub>1</sub>
			MOVE	P <sub>3</sub>	S <sub>3</sub>	P <sub>1</sub>	S <sub>1</sub>
			MOVE	P <sub>4</sub>	S <sub>4</sub>	P <sub>3</sub>	S <sub>3</sub>
			MOVE	P <sub>2</sub>	S <sub>2</sub>	P <sub>4</sub>	S <sub>4</sub>
			MOVE	W	S <sub>1</sub>	P <sub>2</sub>	S <sub>2</sub>
8	4	If I <sub>1</sub> = G <sub>3</sub> and I <sub>2</sub> = G <sub>4</sub> and I <sub>3</sub> = G <sub>2</sub>	MOVE	P <sub>1</sub>	S <sub>1</sub>	W	S <sub>1</sub>
			MOVE	P <sub>4</sub>	S <sub>4</sub>	P <sub>1</sub>	S <sub>1</sub>
			MOVE	P <sub>2</sub>	S <sub>2</sub>	P <sub>4</sub>	S <sub>4</sub>
			MOVE	P <sub>3</sub>	S <sub>3</sub>	P <sub>2</sub>	S <sub>2</sub>
			MOVE	W	S <sub>1</sub>	P <sub>3</sub>	S <sub>3</sub>
9	4	If I <sub>1</sub> = G <sub>2</sub> and I <sub>2</sub> = G <sub>3</sub>	MOVE	P <sub>1</sub>	S <sub>1</sub>	W	S <sub>1</sub>
			MOVE	P <sub>4</sub>	S <sub>4</sub>	P <sub>1</sub>	S <sub>1</sub>
			MOVE	P <sub>3</sub>	S <sub>3</sub>	P <sub>4</sub>	S <sub>4</sub>
			MOVE	P <sub>2</sub>	S <sub>2</sub>	P <sub>3</sub>	S <sub>3</sub>
			MOVE	W	S <sub>1</sub>	P <sub>2</sub>	S <sub>2</sub>
10	4	If I <sub>1</sub> = G <sub>4</sub> and I <sub>4</sub> = G <sub>1</sub>	EXCHANGE	P <sub>1</sub>	S <sub>1</sub>	P <sub>4</sub>	S <sub>4</sub>
			EXCHANGE	P <sub>2</sub>	S <sub>2</sub>	P <sub>3</sub>	S <sub>3</sub>
11	4	If I <sub>1</sub> = G <sub>2</sub> and I <sub>2</sub> = G <sub>1</sub> and I <sub>3</sub> = G <sub>4</sub>	EXCHANGE	P <sub>1</sub>	S <sub>1</sub>	P <sub>4</sub>	S <sub>4</sub>
			MOVE	P <sub>3</sub>	S <sub>3</sub>	W	S <sub>3</sub>
			MOVE	P <sub>4</sub>	S <sub>4</sub>	P <sub>3</sub>	S <sub>3</sub>
			MOVE	W	S <sub>1</sub>	P <sub>4</sub>	S <sub>4</sub>
12	4	If I <sub>1</sub> = G <sub>3</sub> and I <sub>2</sub> = G <sub>4</sub> and I <sub>3</sub> = G <sub>1</sub>	EXCHANGE	P <sub>1</sub>	S <sub>1</sub>	P <sub>3</sub>	S <sub>3</sub>
			MOVE	P <sub>2</sub>	S <sub>2</sub>	W	S <sub>2</sub>
			MOVE	P <sub>4</sub>	S <sub>4</sub>	P <sub>2</sub>	S <sub>2</sub>
			MOVE	W	S <sub>1</sub>	P <sub>4</sub>	S <sub>4</sub>

## 6. 作業実行部

### 6.1 ロボットマニピュレータコマンドの生成

作業実行部では、生成された作業手順をロボットのためのコマンド群に変換される。まず、「MOVE P<sub>i</sub> S<sub>i</sub> P<sub>j</sub> S<sub>j</sub>」(エリアP<sub>i</sub>にある状態S<sub>i</sub>の積木をエリアP<sub>j</sub>に状態S<sub>j</sub>で移動しなさい。)という作業は、以下の様なサブタスクに分けられる。

Step 1. エリアP<sub>i</sub>の上部にマニピュレータの手先を移動。

Step 2. マニピュレータの手先を開く。

Step 3. 積木の状態S<sub>i</sub>に合わせ、マニピュレータ

の手首を回転する。

Step 4. マニピュレータの手先を降下させる。

Step 5. マニピュレータの手先を閉じ、積木を把握する。

Step 6. エリアP<sub>i</sub>の上部にマニピュレータの手先を移動。

Step 7. 移動先のエリアP<sub>j</sub>の上部にマニピュレータの手先を移動。

Step 8. 積木の状態S<sub>j</sub>に合わせ、マニピュレータの手首を回転する。

Step 9. マニピュレータの手先を降下させる。

- Step10. マニピュレータの手先を開き、積木を置く。
- Step11. エリア  $P_j$  の上部にマニピュレータの手先を移動。

次に、分解された11個のサブタスクからロボットマニピュレータコマンドを生成する。本実験で使用するロボットマニピュレータRM101の動作コマンドは、インテリジェントコマンド<sup>(5)</sup>と呼ばれており、全部で15種類用意されている。表4に、主要な11種類のコマンドとそれらの機能を示す。上で述べたサブタスクは、表4の動作コマンドに置き替え、ロボットマニピュレータに与える。

Table 4 Intelligent commands for RM101

コマンド	コマンド機能
H	ロボットの動作基準位置の決定。
I	各関節の回転。
E	現在位置の記憶。
M	指定されたロケーションNo.の位置に移動。
P	ロケーションNo.の指定及び各関節の回転。
N	ロボットの動作基準位置への移動。
C	手先を閉じる。
O	手先を開く。
S	ロボットの駆動速度指定。
T	ロボットの動作の一時停止。
L	各関節の可動範囲をソフト的に指定。

### 6.2 ロボットマニピュレータの手先位置・姿勢制御<sup>(3)</sup>

本節では、幾何学的手法を用いて、4リンク5関節ロボットマニピュレータの逆運動学問題を解き、マニピュレータの手先位置・姿勢制御方式について説明する。まず、本論文で用いる4リンク5関節ロボットマニピュレータの簡略図を図11に示す。ここで、ロボットマニピュレータにより積木を把握させるために、手先位置  $P(x_P, y_P, z_P)$  と手先の水平方向に対する角度  $\alpha$  から、各関節角度  $\theta_1, \dots, \theta_4$  を決定する必要がある。図11において、手先のねじり角  $\beta$  は、手先の位置には影響を及ぼさないが、積木の方向に対応して決定される。

まず、図11より手首関節位置  $R(x_R, y_R, z_R)$  は、手先位置と手先の水平角度から、次式で与えられる。

$$x_R = x_P - l_4 \cos \alpha \cos \theta_1 \tag{6.1}$$

$$y_R = y_P - l_4 \cos \alpha \sin \theta_1 \tag{6.2}$$

$$z_R = z_P - l_4 \sin \alpha \tag{6.3}$$

ロボット座標系の  $x, y$  方向に関する関節角度は  $\theta_1$  のみであるから、 $\theta_1$  は手先位置より次式で与えられる。

$$\theta_1 = \tan^{-1}(y_P/x_P) \tag{6.4}$$

次に、 $\triangle RES$  において余弦定理を用いると、関節角度  $\theta_3$  が以下のように求められる。

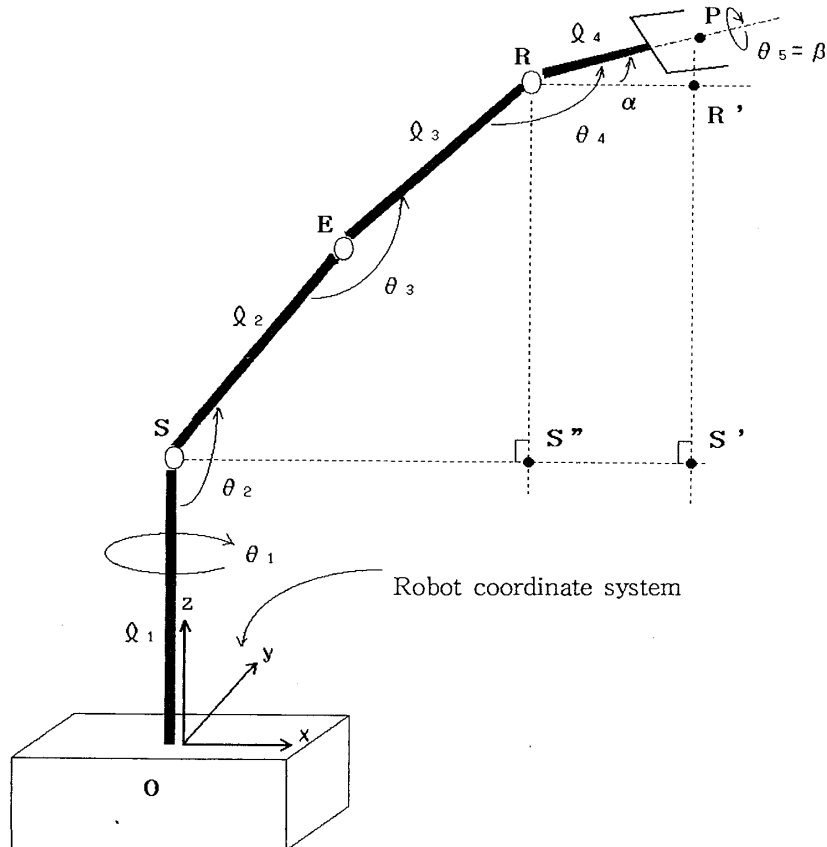


Fig.11 Schematic figure of robot manipulator RM101

$$\theta_3 = \cos^{-1} ((l_2^2 + l_3^2 - L^2) / 2 l_2 l_3) \quad (6.5)$$

但し、

$$L^2 = x_R^2 + y_R^2 + (z_R - l_1)^2 \quad (6.6)$$

また、 $\triangle RSS'$ において、 $\angle RSS'$ は次式で与えられるので、関節角度 $\theta_2$ は、(6.8)式で求められる。

$$\angle RSS' = \sin^{-1} ((z_R - l_1) / L) \quad (6.7)$$

$$\begin{aligned} \theta_2 &= \angle ESR + \angle RSS' + \pi / 2 \\ &= \theta_3 + \sin^{-1} ((z_R - l_1) / L) + \pi / 2 \end{aligned} \quad (6.8)$$

さらに、5角形 $R'ESS'$ の内角の和が次式で与えられることから、

$$\angle R'RE + \angle RES + \angle ESS' + \pi = 6\pi \quad (6.9)$$

$\angle R'RE$ が関節角度 $\theta_2$ 、 $\theta_3$ から(6.10)式で与えられ、

$$\angle R'RE = \pi / 2 - \theta_3 - \theta_2 \quad (6.10)$$

最後に、関節角度 $\theta_4$ が次式より求められる。

$$\theta_4 = \angle R'RE + \alpha = \pi / 2 - \theta_3 - \theta_2 + \alpha \quad (6.11)$$

手先位置 $P(x_P, y_P, z_P)$ と手先の水平方向に対する角度 $\alpha$ から、各関節角度 $\theta_1, \dots, \theta_4$ を決定する為のアルゴリズムをまとめると以下の様に与えられる。

$$\theta_1 = \tan^{-1} (y_P / x_P) \quad (6.4)$$

$$\theta_3 = \cos^{-1} ((l_2^2 + l_3^2 - L^2) / 2 l_2 l_3) \quad (6.5)$$

$$\theta_2 = \theta_3 + \sin^{-1} ((z_R - l_1) / L) + \pi / 2 \quad (6.8)$$

$$\theta_4 = \pi / 2 - \theta_2 - \theta_3 + \alpha \quad (6.11)$$

$$L^2 = x_R^2 + y_R^2 + (z_R - l_1)^2 \quad (6.6)$$

ここで、 $l_1, l_2, l_3, l_4$ はロボットマニピュレータの各リンク長を表す。また、点 $R, E, S$ の座標を各々、 $(x_R, y_R), (x_E, y_E), (x_S, y_S)$ とする。但し、各リンク長、各関節角度の物理的制約から、手先位置の動作範囲が限定され、以下の様な場合、手先が目的とする位置に到達することが不可能となる。

- (1)  $L' > l_2 + l_3 + l_4$   
但し、 $L' = \sqrt{x_H^2 + y_H^2 + (z_H - l_1)^2}$
- (2)  $L > l_2 + l_3$

## 7. 積木並び替え作業実験

前節までに提案した作業制御システムを用いた積木並び替え作業の実験を行う。ここで、積木A, B, C, D (初期状態)の並びをD, A, B, C (目標状態)に並び替える一連の作業の様子をFig. 12(a)~(o)に示す。順に、積木Aのワークエリアへの移動、積木Dの移動、積木Cの移動、積木Bの移動、積木Aの移動を示している。また、Fig. 12(a)~(o)の作業において初期状態を認識し、目標状態を達成するまでの処理時間は約300秒を要した。これは、マニピュレータの手先移動速度が7 [cm/秒]と非常に遅いためである。さらに、初期状態A, B, C, Dから24通りの目標状態への並び替え作業実験を行った結果、積木の認識処理、作業手順生成処理については24通りの場合について、全て良好な結果を得た。

しかし、Block Dを移動する際、手先から滑り落ちるケースが5通り程度、発生した。これは、手先を回転する際、マニピュレータ機構上の原因で手先の握力が緩む為である。今後、このような作業エラーに対して、触覚センサーなどの利用によるエラー検出、リカバリー作業の生成が必要と思われる。さらに、積木を把握する際の手先位置誤差が最大 $\pm 5$  [mm]であり、複雑な組立作業への適用に際しては、カメラ座標系からロボット座標系への変換精度の向上が必要である。

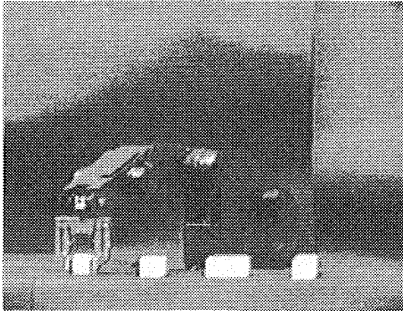
## 8. 結言

本論文では、視覚情報を利用したロボットマニピュレータ作業制御システムを提案し、積木の並び替え問題に適用し、実験によりその有効性を示した。まず、視覚情報処理部では、2台のCCDカメラを用いた積木の位置、方向、種類及び、積木間の相対関係等の認識手法を提案し、積木の認識実験によりその機能を確認した。本認識手法では、対象物が柱状物体であることを積極的に利用した簡便な方法であり、対象物の個数が増加しても、有効であることが推測できる。次に、作業計画部では、マクロ言語で記述された作業手順データベースを用いて、積木を初期状態から目標状態に並び替えるための手順の生成方法について述べた。ここで、開発したマクロ言語は汎用性に富んでおり、他の組立作業などにも利用可能である。さらに、作業実行部では、積木の並び替え作業手順をサブタスクに分け、次に、ロボットコマンド群に変換する手法を示した。多様なサブタスク群を開発することにより、さらに複雑な作業を実行することが可能である。最後に、提案した作業制御システムを用いて、4個の積木の並び替え作業実験を24通りの組合せに対して実施し、本システムの有効性を確認した。なお、本実験では、積木の位置検出に対して数mmの誤差を許容しても、積木の並び替え作業が実行可能であったが、組立作業等の複雑な作業環境問題に対しては、高精度な位置検出手法の開発が必要である。また、アルゴリズムで表現不可能な作業に対しても、適応可能な学習機能を持つ制御手法、さらに、作業中の誤動作の自動検出、自動修復等の制御手法の開発を進める予定である。

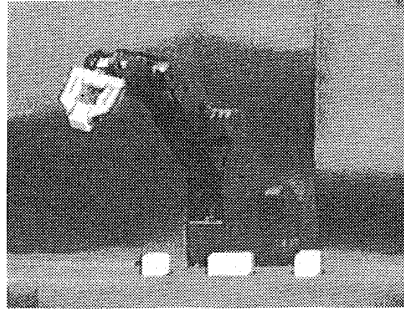
## 参考文献

- (1) 池井他, 「センサ情報に基づいた2台のロボットの協調制御システムの開発」, 日本ロボット学会誌, Vol.8, No.3, pp.265-272, 1990.
- (2) 長田, 「ロボット工学—知能情報編—」, pp.176-202, 昭晃堂, 1986.
- (3) 廣田, 「あいまい制御知能ロボット」, pp.171-177, マグロウヒルブック, 1985.

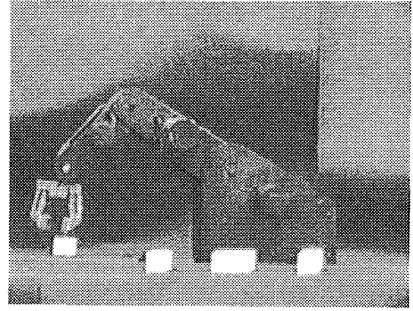
- (4) 田村, 「コンピュータ画像処理入門」, pp.93-94,  
総研出版, 1985.  
(5) 三菱電機株, 「ムーブマスターRM101取扱説明書」,  
pp.7-10, 1982.



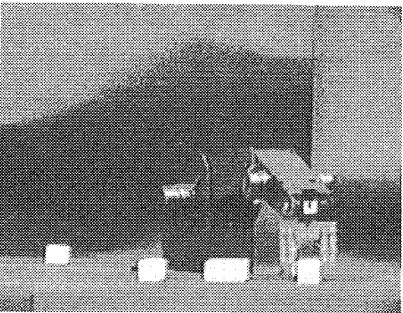
(a) Motion of arms picking block A



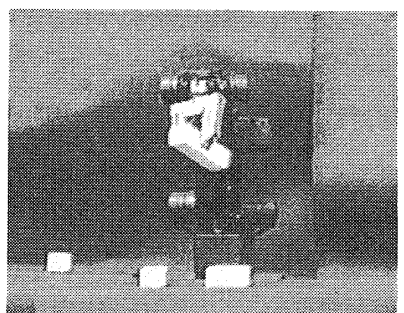
(b) Motion of arms carrying block A



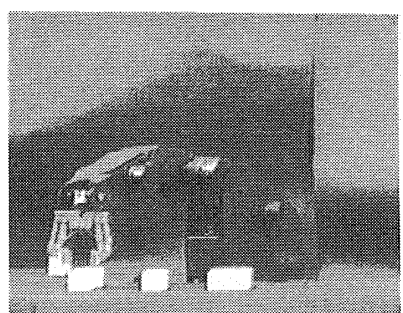
(c) Motion of arms putting block A



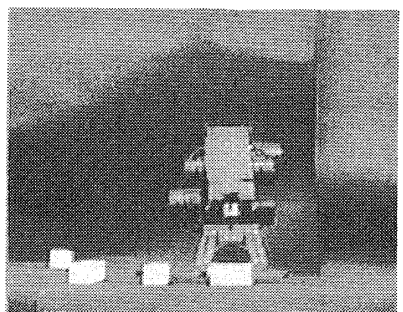
(d) Motion of arms picking block D



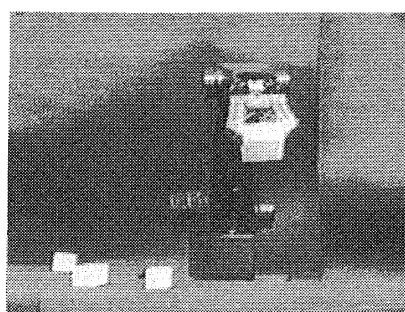
(e) Motion of arms carrying block D



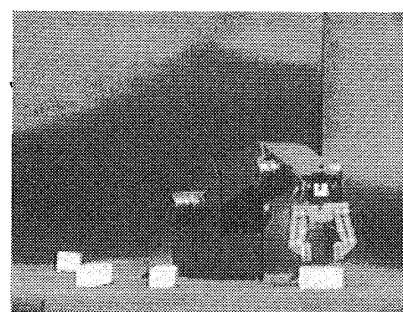
(f) Motion of arms putting block D



(g) Motion of arms picking block C

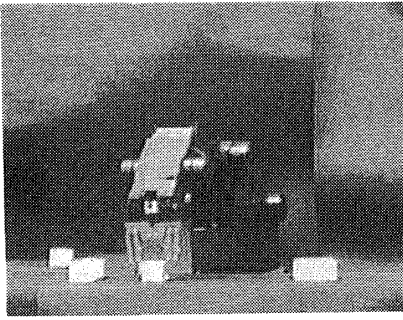


(h) Motion of arms carrying block C

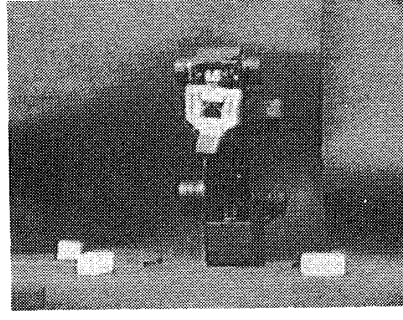


(i) Motion of arms putting block C

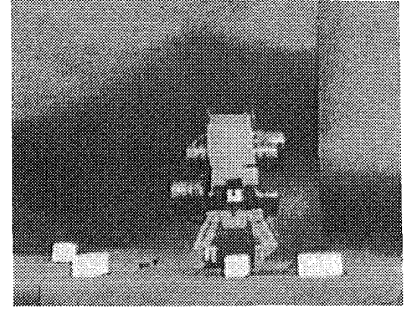
Fig.12 Photos doing a series of rearrangement tasks



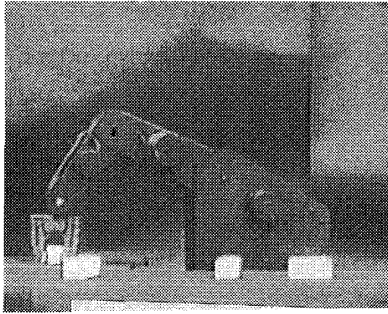
(j) Motion of arms picking block B



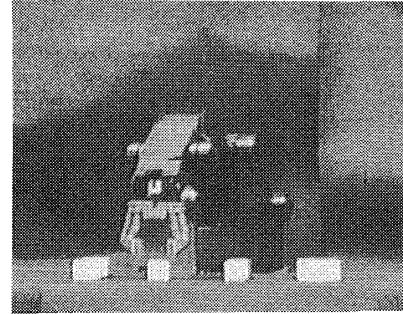
(k) Motion of arms carrying block B



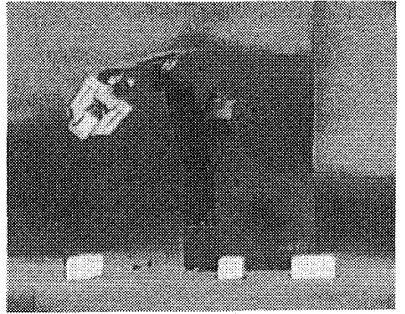
(l) Motion of arms putting block B



(m) Motion of arms picking block A



(n) Motion of arms carrying block A



(o) Motion of arms putting block A