

ロボットによる紐のハンドリング

稻葉 雅幸* 井上 博允**

本論文は、ロボットにより紐を操作する際の視覚・行動系の協調動作に関する研究について述べたものである。紐のように柔構造の物体を操作するには視覚情報の利用が必要不可欠である。本研究では視覚情報を用いて動作を修正し、動作結果をもとに注視する領域を制御し動作確認をするという協調動作の作業例として紐のハンドリングをとりあげる。

本論文では、視覚・行動系の協調動作の実現方法を解析し、紐のハンドリング実験に用いたハンド・アイ・システムと視覚処理アルゴリズムについて述べる。協調動作を実現するための手法として、局所領域を注視する視覚機能、座標系間の較正、視覚による動作確認、視覚による操作量の修正法について述べる。紐の操作実験は紐をつかんでリングに通したあと、紐をつかみ直して結ぶという一連の動作からなる。実験システムは、両眼立体視システム、6+1自由度アームシステム、作業記述用のLispシステムからなる。線分領域の画像処理を組合せて対象を認識し、両眼立体視によって3次元計測を行ない環境の認識と動作の制御を実現している。

視覚により得られる情報を基に操作量を修正し、動作の確認を視覚により行なうことによって、柔軟な紐を3次元空間で操作することができるということを実験により確認した。

1. はじめに

紐のハンドリングは、一見単純なように見えるが、本質的に視覚機能がなくては実現が非常に困難な作業である。なぜならば、紐のような柔軟な物体は定まった形を持たず、操作中にその形状が変化してしまうためである。それゆえ、視覚によって計画動作を修正、確認することが不可欠となる。

本論文で述べる紐のハンドリング実験は、紐をリングに通す作業と紐を結ぶという2つの作業を含んでいる。まず、ロボットは視覚により紐を探し出し、それを把持する。把持したことを視覚で確認し、紐の先端部の長さを計測する。次に、リングの中心位置を計測し、ビジュアル・フィードバック制御に従って紐をリングの中へ誘導していく。通し終ったことを視覚により確認し、紐の先端の位置を計測する。それから、ロボットは紐をリングの反対側から把み直し、視覚により把握を確認したあと紐をフィーダから引き出す。そして最後に、ロボットは紐をリングといっしょに結ぶ。

ここでの視覚の役割は対象物の発見、動作の確認、ビジュアル・フィードバックによるハンドの誘導の3つである。実験ではスキャン・ライン上の特徴抽出を組み合わせて対象を認識し、両眼立体視によって3次元計測を行なっている。そして、得られた計測値をもとに計画動作を修正し、動作後の状況を注視することによって動作確認と次の動作の計画を行なっている。また、協調動作の実現に不可欠なシステムのキャリブレーションにも視覚を利用している。

以下第2章で、視覚・行動系の協調動作の実現方法の解析を行ない、第3章で実験に用いたハンド・アイ・システムの説明、第4章で紐のハンドリング実験について述べる。

2. 視覚・行動系の協調制御の実現

物体の位置決め動作等で視覚と行動系を協調させて制御する場合には、必要な部分のみを注視し動作を計画修正することが多い。そのため、注視する領域を自由に設定しその領域から必要な特徴を抽出する局所画像処理機能が重要になる。注視領域をどのように設定し、どのような特徴を抽出するかが視覚行動系の協調動作実現の際の大きな問題となる。また、3次元空間で動くマニピュ

原稿受付 1985年7月11日

* 東京大学大学院 ** 東京大学

レータを視覚からの情報により制御するには、3次元情報を視覚により直接得られると便利なことが多い。そのため、3次元情報を計測するための手段を持ち合わせていることがハンド・アイ・システム¹⁾にとって重要なことであると思われる。

視覚・行動系の協調制御には、制御情報の流れによって2種類の形式がある。1つは、視覚がハンドの動作を誘導するビジュアル・フィードバックの場合である。もう1つは、動作によって発生する状態の変化を予測することにより、行動系から視覚の注視点が制御されるという場合である。このような双方向の情報変換を可能にするには視覚座標系とマニピュレータ座標系の変換方法を知ることが必要となる。この変換を表わす関数を自動的に校正できることは非常に重要であり、ロボットの自律化の重要な条件であると考える。

視覚系と行動系の相互の座標変換ができると、次は視覚系及び行動系から得られる情報をいかに利用して物体操作の修正と確認を行なうかが問題となる。すなわち、視覚により得られる情報から行動系に送るべき次の操作指令をいかに生成するか、そして、マニピュレータの動きによって引き起こされる操作対象及びハンド部の状況変化の情報をいかに次の認識過程へ利用するか、という2つのことを考察する必要がある。

本研究では上記の問題点に対して、(1)スキャン・ラインにそった特徴抽出、(2)3次元位置を計測する両眼立体視、(3)視覚によりハンドの3次元位置を直接計測することによる変換行列の校正、(4)ハンドの位置を利用した注視点制御、(5)目標空間に注視部が存在するよう動作を修正する物体の誘導法、という方法をとって

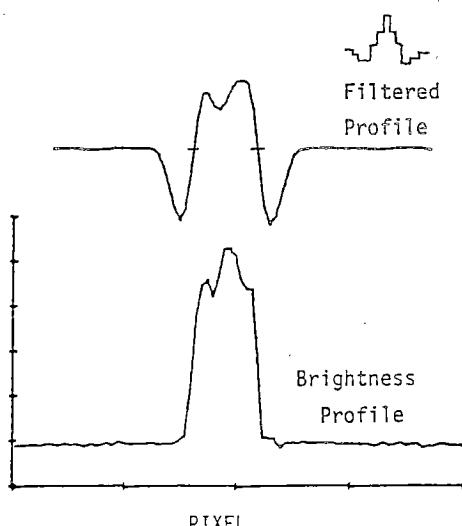


Fig. 1 Detecting zero-crossings along a scan-line

視覚行動系の協調動作を実現している。以下、それについて説明する。

2.1 スキャン・ライン上の特徴抽出

画像処理する領域を1次元の領域とし、その領域をスキャン・ラインと呼ぶ。スキャン・ラインの位置・方向・長さを制御して画像から対象の特徴を抽出する。

Fig.1 にスキャン・ラインにそった明るさの変化と、平均化した2次の差分処理の結果を示す。符号が反転する部分のゼロ点（ゼロクロス）が明るさの急変する点とみなせる。このゼロクロスの位置と、ゼロクロス間の距離と縁点近傍の平均明るさの情報を、注視する対象の探索と位置情報の計算に利用する。対象物体に対してどのようにスキャン・ラインを設定して特徴を抽出するかという制御方法が重要な問題となる。物体の境界線では明るさが急変する場合が多いので、境界線の追跡処理の際にスキャン・ラインの方向を修正しつつ縁点の検出を行なえばよい。また、ゼロクロス間の平均明るさと距離を利用して対象物のしめる領域を切りだすこともできる²⁾。

2.2 両眼立体視

局所処理によって得られた画像上の位置を基に3次元位置を計算するために両眼立体視³⁾の手法を用いる。両眼立体視の場合には左右の画像から対応点を得ることが大きな問題となる。ここでは、対象の形状の特徴的な部分を両画像から探し出すことによって対応を取ることにする。対応点からの距離計算は以下のようになる。

左眼と右眼の視覚基準座標系における方向行列と位置ベクトルをそれぞれ、 L_{ROT} , L , R_{ROT} , R とする。視野内にある点Vの左眼及び右眼からの距離をそれぞれ、 d_l , d_r と表す(**Fig.2** 参照)。左眼及び右眼からVへの方向ベクトルをそれぞれ u_l , u_r とすると、左眼からVへのベクトル V_l 、右眼からVへのベクトル V_r はそれぞれ、

$$V_l = d_l \cdot u_l, \quad V_r = d_r \cdot u_r$$

となる。ここで理論的には、

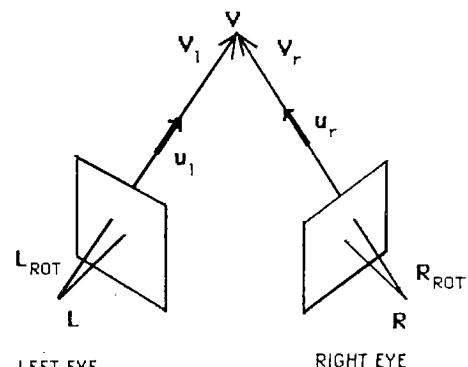


Fig. 2 Geometry for stereopsis

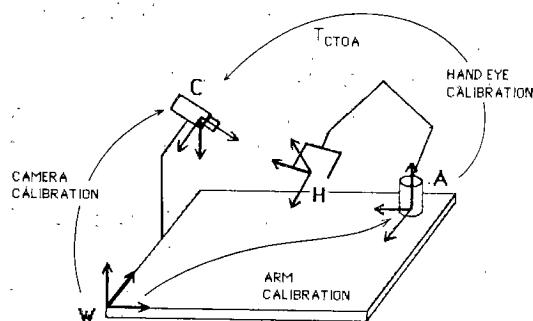


Fig. 3 Arm calibration, camera calibration and hand eye calibration

$$\mathbf{V} = \mathbf{L}_{\text{ROT}} \cdot \mathbf{V}_t + \mathbf{L} = \mathbf{R}_{\text{ROT}} \cdot \mathbf{V}_r + \mathbf{R}$$

が成立するはずであるが、計測誤差又は較正誤差により実際には成立しない。そこで、 d_t, d_r を変数とする評価関数 J を次のように定める。

$$J(d_t, d_r) = |(\mathbf{L}_{\text{ROT}} \cdot \mathbf{V}_t + \mathbf{L}) - (\mathbf{R}_{\text{ROT}} \cdot \mathbf{V}_r + \mathbf{R})|^2$$

J を最小にするように、 d_t, d_r を求めると以下のようになる。

$$d_t = K \cdot ((\mathbf{U}_t, \mathbf{D}) - (\mathbf{U}_t, \mathbf{U}_r) \cdot (\mathbf{U}_r, \mathbf{D}))$$

$$d_r = K \cdot ((\mathbf{U}_t, \mathbf{U}_r) \cdot (\mathbf{U}_t, \mathbf{D}) - (\mathbf{U}_r, \mathbf{D}))$$

$$K = \frac{1}{1 - (\mathbf{U}_t, \mathbf{U}_r)^2}$$

ただし、 $\mathbf{U}_t = \mathbf{L}_{\text{ROT}} \cdot \mathbf{u}_t$, $\mathbf{U}_r = \mathbf{R}_{\text{ROT}} \cdot \mathbf{u}_r$, $\mathbf{D} = \mathbf{R} - \mathbf{L}$

視覚基準座標系における点 V の座標は、次の式で求めることができる。

$$\mathbf{V} = \frac{1}{2}(d_t \cdot \mathbf{U}_t + d_r \cdot \mathbf{U}_r) + \frac{1}{2}(\mathbf{L} + \mathbf{R})$$

\mathbf{u}_t と \mathbf{u}_r は左右画像中の対応点の位置とカメラの内部パラメタから求めることになる。

2.3 ハンド・アイ・キャリブレーション

ハンド・アイ・システムの較正すべきパラメタとしては、(1) マニピュレータの構造パラメタ、(2) カメラの内部パラメタ、(3) マニピュレータとカメラの相対位置と方向、がある。(1), (2) については変化することが少ないため一度較正してしまえばよいが、(3) は作業に合わせて変化させることがあるために、較正方法が簡便であることが望ましい。

ここでは、この較正を自動的に行なうために、任意の場所にあるマニピュレータの位置を視覚により計測し、ジョイント角から求めた座標値と対応をとることによって座標変換に必要な情報を得るという方法を用いる⁴⁾。本方法の利点は較正に用いた計測系を作業の実行時にも利用するところにあり、計測系の絶対精度が少々悪くてもその誤差を補償することができる。また、対応をとるためにサンプリング空間を適当に選ぶことによって作業

場所に合わせた変換式を用いることができ、特に視覚を積極的に移動して利用する場合に効果があると考えられる。そして、作業実行時に較正することもできるため、視覚とマニピュレータの相対位置が作業中に変化した場合にも適応的に対処することが可能となる⁵⁾。

以下、ハンドの位置のみを視覚で計測して、座標変換行列を得る方法の概要を示す。Fig. 3 に、ハンド・アイ・システムの座標系モデルを示す。 W, A, H, C はそれぞれ世界座標系、マニピュレータ基準座標系、ハンド座標系、視覚装置基準座標系をさす。

ハンド座標系の原点の位置 H を、視覚座標系(C)における位置ベクトル \mathbf{h}_C と、アーム座標系(A)における位置ベクトル \mathbf{h}_A で表す。 A と C の座標変換を表わす式を同次座標表現を用いて次のように定義する。

$$\mathbf{h}_A = T_{C \rightarrow A} \cdot \mathbf{h}_C$$

空間内の複数の点へハンドを移動し、第 i 番目の計測点における

$$\mathbf{h}_{Ai} = (x_{Ai}, y_{Ai}, z_{Ai}, 1)^T \text{ 及び}$$

$$\mathbf{h}_{Ci} = (x_{Ci}, y_{Ci}, z_{Ci}, 1)^T$$

を得る。 \mathbf{h}_{Ai} はマニピュレータのジョイント角から求め、 \mathbf{h}_{Ci} は視覚により得るものとする。

理論的には 3 点のサンプルで $T_{C \rightarrow A}$ は決まるが計測誤差を少なくするために x, y, z 座標ごとの最小自乗法を用いて求めるとすると

$$T_{C \rightarrow A} = H_A \cdot H_C^T \cdot (H_C \cdot H_C^T)^{-1}$$

$$H_A = [\mathbf{h}_{A1}, \mathbf{h}_{A2}, \dots, \mathbf{h}_{An}]$$

$$H_C = [\mathbf{h}_{C1}, \mathbf{h}_{C2}, \dots, \mathbf{h}_{Cn}]$$

$$\text{ただし, } n \geq 4$$

となる。

ここでサンプルした H_A, H_C を蓄えておけば再較正する場合に利用できる。すなわち、相対位置がそれほど変化していない場合に、蓄えてある \mathbf{h}_{Ai} の位置へハンドを移動し、ハンドが視野内に入っているかどうかを \mathbf{h}_{Ci} をを利用して認識する。視野内にあれば \mathbf{h}_{Ci} の近傍からハンドを探しだし位置を求め、なければ次の $\mathbf{h}_{A(i+1)}$ へハンドを移動する。この操作をプログラムで実行することにより、簡単な自動キャリブレーションが実現できる。

2.4 視覚による動作の確認

一連の動作からなる作業を実現するには、各動作の確認を行なわないと以後の作業に支障をきたすことがある。特に把握動作の場合には、把握が成功していないと次の操作は無意味になる。また、把握した対象のハンドに対する相対位置関係を知らなくては、次の動作におけるハンドの目標位置・姿勢をどこに設定して良いかわからない。そのため、把握動作を確認し、把握した対象物の形

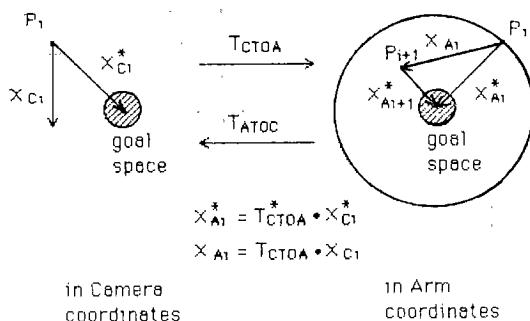


Fig.4 Error vectors at visual feedback operation

状を再認識することが必要になる。この確認処理を視覚によって実行する場合、ハンドの位置・姿勢を視覚系へフィードバックすることによって、注視領域の制御が可能となる⁶⁾。すなわち、ハンドと把握物体が視覚画面のどの部分にどういうふうに見えるかを画面への投影計算から予測することによって、注視する領域の設定と特徴抽出処理の選択を行なうことができる。この投影計算の

際に 2.3 で述べた座標変換式が必要となり、局所特徴抽出機能を用いて対象を再認識するためにハンドと対象の形状情報を利用する。この際、対象の視覚に対する向きを変えたり、視点を変えるという操作を行なえれば、認識処理を助けることもできる。

2.5 ビジュアル・フィードバック

視覚を用いて把握対象物を誘導するいわゆるビジュアル・フィードバック動作⁷⁾では、対象の位置の計測結果をもとにハンドの操作量を適宜修正することになる。このほか、対象の計画軌道と対象の位置計測とからハンドの修正軌道を生成することが必要となるが、現在値と目標値の差ベクトル分の移動をそのまま指令すると行きすぎたり方向があやまっていたりして環境を破壊することがある。そこで、計画軌道に沿って目標位置を少しづつ移動させ、その目標位置の近傍に対象の注視部が存在するようにビジュアル・フィードバックをかけるという方法をとる。このようにする場合には、注視部に対する目標位置は大きさを有する空間であることが動作の効率

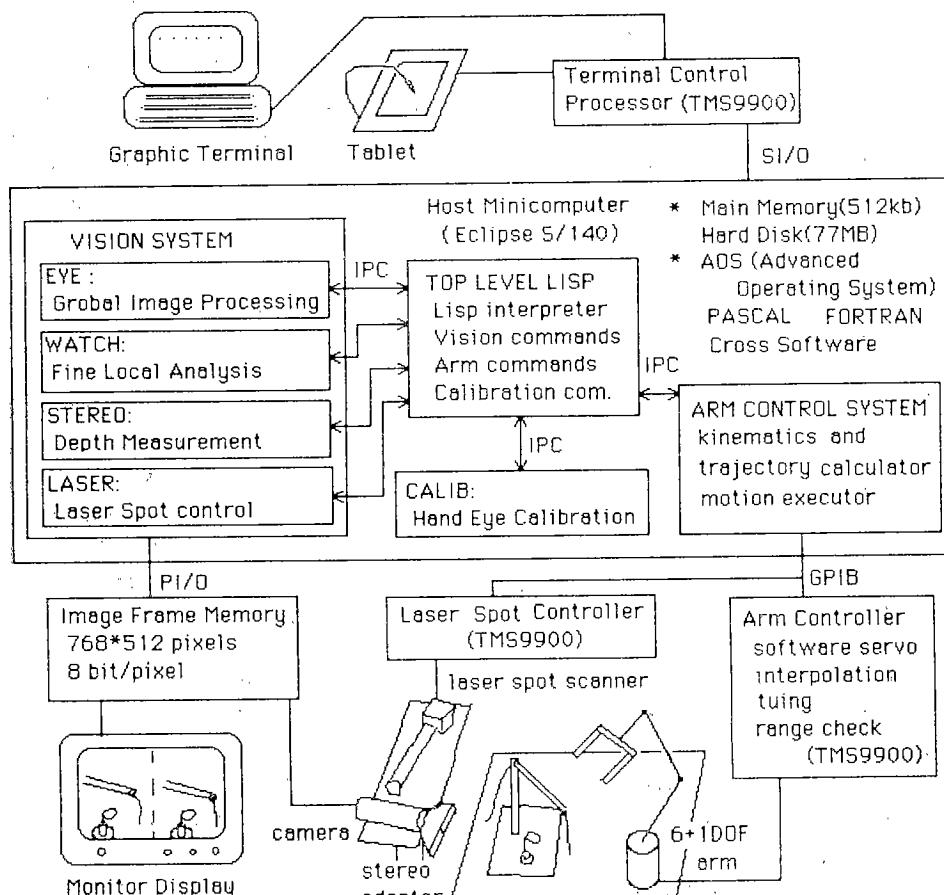


Fig.5 Hand eye system configuration

Table 1 Basic commands for the experiment

<u>Arithmetic functions</u>		
(!* coordinates ...)	(V+ pos ...)	(IN-SPACE vector space)
(DIVIDE position1 position2 scale)		(LINE-CENTER line)
(CREATE-POS pos distance angle)		
<u>Arm command</u>		
(MOVE locations)	(RELMOVE dx dy dz)	(RELVMOVE vector)
(REACH distance)	(SWEEP distance)	(LIFT distance)
(ROLL angle)	(PITCH angle)	(YAW angle)
(CLOSE-HAND force)	(OPEN-HAND width)	(VELOCITY {value})
<u>Vision command</u>		
(SCAN-LINE pos angle length threshold)		
(FIND-EDGES pos width threshold)		(STILL)
(STEREO-DIST left-pos right-pos)		(STEREO-PROJECT vector)
<u>Calibration command</u>		
(SET-CALIBDATUM vector1 vector2)		(SET-DATACOUNT number)
(RD-CALIBDATA)		(RD-DATACOUNT)
(CALIBRATE rowsize columnsiz)		
(EYETOARM coordinates! vector)		(ARMOEYE coordinates! vector)
<u>Display command</u>		
(DRAWTEXT pos text scale)		(SHOWLINE startpos endpos)
<u>For rope handling</u>		
(GRASP-ROPE coordinates! vector)		(RELEASE-ROPE)
(FIND-ROPE pos width brightness)		(VERIFY-ROPE pos)
(TRACK-ROPE-END pos direction length)		
(FIND-RING pos width brightness)		
(FIND-RING-CENTER position length)		
(SELECT-RING edgels)		(SELECT-ROPE edgels)
(IN-RINGSPACE vector bottom top)		(IN-WORKSPACE vector)
(VECTOR-DIVIDE vector scale1 scale2)		
(DISP-MESSAGE messages)		

を上げるうえで望ましい。そのため、本研究では視覚により対象物の誘導が正しく実行されているかどうかの判定をするための目標空間を対象にあわせて設定し、その目標空間を常時更新することによって対象を誘導する方法をとった。

対象物の誘導は、この目標空間内に対象物の注視部分が存在するかどうかを視覚により判定し、存在しなければ偏差量からハンドへの操作量を計算して行動系へ送り、存在していれば新しい目標空間を設定するという一連の処理を繰り返すことによって実現できる。視覚座標で得られる偏差量をもとにハンドの操作量を計算するために2.3で得た変換式を用いる。

この場合には変換行列の方向成分のみが関係することになり、キャリプレーションの精度がフィードバックの収束性に影響する。変換行列と修正動作の収束性の関係をみるためにFig.4を示す。 x_{Ai} は*i*回の修正後の偏差量を視覚座標系で計測したものであり、 x_{At} はアーム座標系での偏差量であり $x_{At} = T_{CtoA} \cdot x_{Ci}$ により求める。上付記号*で真の値を示すことにして、

$$\begin{aligned} x_{A(i+1)}^* &= x_{Ai}^* - x_{Ai} \\ &= T_{CtoA}^* \cdot x_{Ci}^* - T_{CtoA} \cdot x_{Ci} \\ \text{視覚の計測値が正しいとして } x_{Ci} &= x_{Ci}^* \text{ とみなせば,} \\ x_{A(i+1)}^* &= (T_{CtoA}^* - T_{CtoA}) \cdot x_{Ci}^* \\ &= (T_{CtoA}^* - T_{CtoA}) \cdot T_{Atoc}^* \cdot x_{Ai}^* \\ &= (I - T_{CtoA} \cdot T_{Atoc}) \cdot x_{Ai}^* \end{aligned}$$

よって、 $D = I - T_{CtoA} \cdot T_{Atoc}^*$ と置けば、 $x_{An}^* = D^n \cdot x_{A0}^*$ となる。 n が大きくなつた時 $|x_{An}^*|$ が小さくなるためには D の固有値の絶対値がすべて 1 より小さければよい。

3. ハンド・アイ・システムの構成

実験に用いたハンド・アイ・システムの構成をFig.5に示す。本システムは、知能ロボット・プログラミングシステム COSMOS^⑤の基本モジュールをハンド・アイ実験用に再構成したものである。

アームは把握のための 1 自由度を含めて 7 自由度を有し、各関節のサーボは 1 台の 16 ビットマイクロコンピュータ上のプログラムで実現されている。カメラからの画像データはパラレル・ポートを

通してミニコンピュータと接続しているイメージ・フレーム・メモリー (IFM) に蓄えられる。視覚の補助装置として、レーザ・ビームの方向を制御するレーザ・スポット・スキャナがある。中心となる計算機は、データゼネラル社の 16 ビットミニコン Eclipse S/140 である。

ミニ・コンピュータ内のソフトウェアは、Top Level Lisp と、Calibration, Vision, Arm のサブシステムからなる。Top Level Lisp はロボットの作業プログラムを記述するためのモジュールであり、他のサブシステムへ命令を発する基本関数を有する。視覚系と行動系の座標系間の変換行列を較正するためのモジュールが CALIB であり、計測データ及び較正結果の評価と再利用のための手続きを含む。Vision subsystem は機能により、EYE, WATCH, STEREO, LASER の 4 つのモジュールに分かれており^⑩、Arm はジョイント角を計算するためにある。

視覚モジュール EYE は、シーンの大まかな解析を広い領域に対して行うためのもので、64×64 の大きさをもつ汎用ウインドウを通して、空間微分等の画像演算、細線化、論理演算、エッジ抽出、統計解析等の処理プログラムを有する。WATCH は局所領域を精密に解析し、効率的な画像認識を実現するためのモジュールであり、位置、方向、長さを自由に設定できるスキャン・ラインと、矩形の小ウインドウを有する。STEREO と LASER が 3 次元位置を計測するためのモジュールである。局所

領域の濃淡の相関と、エッジの位置、方向、強さの相関を用いて対応点を探索する方法、レーザ・ビームを投影しレーザ・スポットの位置から対応点を得る方法を利用することができる。

各モジュールはミニコン上 のマルチ・プロセス OS (AOS) の下でプロセスとして管理され、プロセス間通信機能 (IPC) を用いて交信している。

4. プログラミングと実験

ハンド・アイ・システムの諸機能を利用するため、Lisp から各サブシステムへ指令を発するためのシステム基本命令があり、それらを組みあわせて紐の操作用プログラムを記述している。Table 1 に Arm 駆動、視覚情報抽出、キャリブレーション、及び、表示用の主なシステム基本命令と、本実験用に用意した主な命令を示す。動作指示には 3 次元の同次座標表現を用い、相対移動命令を充実させている。視覚用には 1 次元の領域から対象の縁点を抽出するための関数と、縁点データを操作するための諸関数を整備している。

Fig. 6 に紐をリングに通したあと結ぶという実験プログラムの概略フローを示す。番号が四角で囲まれているステップは、マニピュレータの操作を行なうことを示し、丸で囲まれているものは視覚の処理を行なうことを示す。

まず、ロボットは視覚により紐を捜し出し、紐の端点とフィーダの先端の位置を計測する。その 2 点からグラスプ・ポイントを計算し、紐をつかむ。把持したことを視覚で確認するためにハンドをすこし移動し以前の位置に紐がないことを検査する。うまく把持されていれば、紐の先端部の長さを計測するための処理を行なう。

次に、ロボットはリングの中心位置を計測したあと、把持部の紐の長さを考慮してリングの上方へ紐を移動する。そしてビジュアル・フィードバック制御に従って紐をリングの中へ誘導してゆく。通し終ったことを視覚により確認したあと紐を離し、紐の先端の位置を計測する。

リングの反対側から紐を把み直すためのグラスプ・ポイントを紐とリングの位置から計算し、把むだけの余裕があれば把握動作にはいる。確実に把握しているかどうかを調べるために紐を手前に引いて把握動作の確認をする。

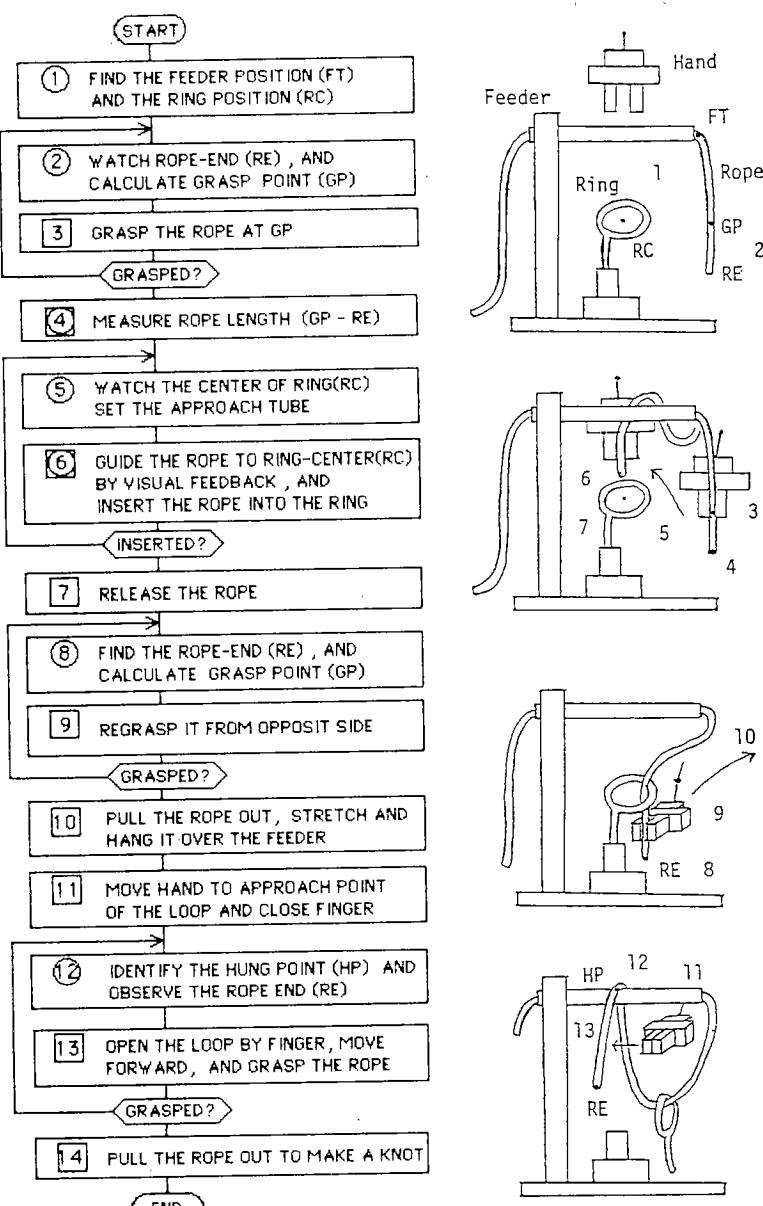


Fig. 6 Flow Diagram of rope-into-ring and rope-tying operation

る。視覚により把握を確認したあと紐を引き出してリングを外し、フィーダに紐を掛ける。

フィーダの先端の位置を基に掛けられた紐の位置を計測し指のアプローチ・ポイントを決める。つかみやすいように指を閉じて紐に近づき、指を開くことによって手前の紐をずらし、紐の位置を再確認してから把持する。把持した紐を引き出し、フィーダから外すことによってリングを紐で結ぶ作業が完結する。

視覚処理と動作の制御アルゴリズムを次に説明する。

4.1 ハンド・アイ・キャリブレーション

視覚で探索しやすいようにキャリブレーション用の白い球を持たせたハンドを紐の操作空間内で移動し、両眼立体視によりハンドの位置を計測する。サンプルする空間は視野内のうち特に紐を最初に把握する場所の近傍を重点的にサンプルし変換行列を得る。較正結果を基にハンドの位置を視覚画面に表示し、結果が十分であるかどうかの検査を行なうことによってサンプリングの打切りを決める。

4.2 紐の発見と追跡

Fig. 7 に紐の端点の探索過程におけるスキャン・ラインの移動例を示す。まず最初に水平スキャン・ラインで縁点を検出し、縁点間の距離と平均明るさをあらかじめ与えてある紐のパラメタと比較して紐の候補を得る。次に、小さなウインドウを候補の中心に設定し、ウインド

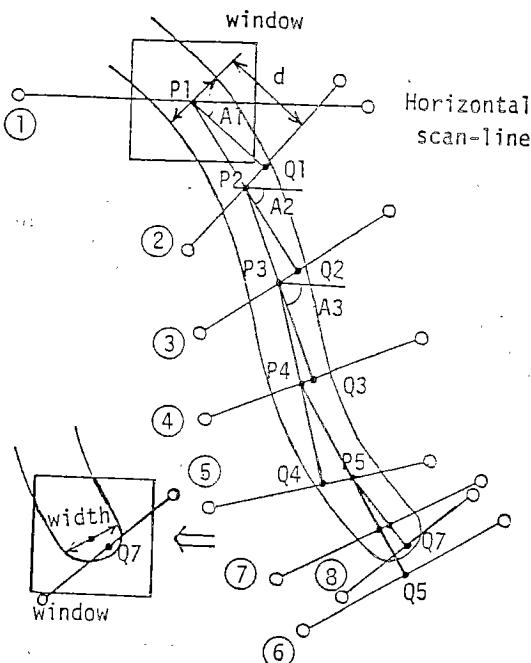


Fig. 7 Tracking a rope and detecting the rope-end

ウ内の線画抽出を行なう。もし、一対の平行線がありその間隔が紐の太さから予測できる値に近い場合に、その候補を紐とみなす。紐の中心線の正確な位置と方向、平均明るさ及び巾を計算し、この後の追跡処理に利用する。

スキャン・ラインの位置、方向を次式のように制御している。

$$Q_i = P_i + d \cdot (\hat{x} \cdot \cos(A_i) + \hat{y} \cdot \sin(A_i))$$

$$A_{(i+1)} = \cos^{-1}\left(\frac{\mathbf{P}_{(i+1)} - \mathbf{P}_i}{|\mathbf{P}_{(i+1)} - \mathbf{P}_i|}\right)$$

ただし、 d は外挿のための定数、 \mathbf{P}_i と A_i はそれぞれ紐の第 i 番目の中心位置と方向角、 \hat{x} と \hat{y} は画像座標の基本単位ベクトルである。スキャン・ラインにそった紐の中心位置検出を行なって \mathbf{P}_i を得ることにより、次の中心位置の予測値である $\mathbf{Q}_{(i+1)}$ を上式により決める。

Fig. 7 のスキャン・ライン 6 のように、 \mathbf{P}_i の検出が失敗すると、点 $\mathbf{Q}_{(i-1)}$ と点 $\mathbf{P}_{(i-1)}$ の間を 2 分法により紐の端点検出を行なう。

4.3 リングの中心位置の認識

リングは画像中で橢円形の形状を持つ領域として映る。初期状態においてリングの位置を予測できない場合には、紐と同様の手法を用いてリングを探索する。リン

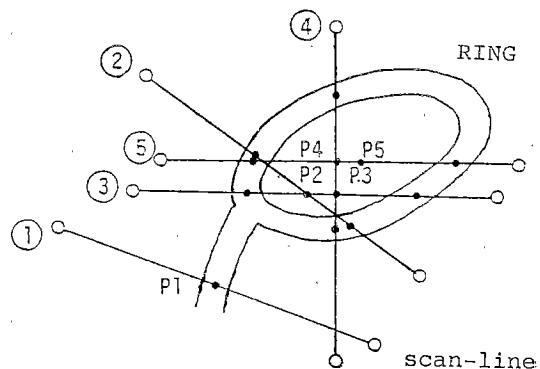


Fig. 8 Finding the center of a ring

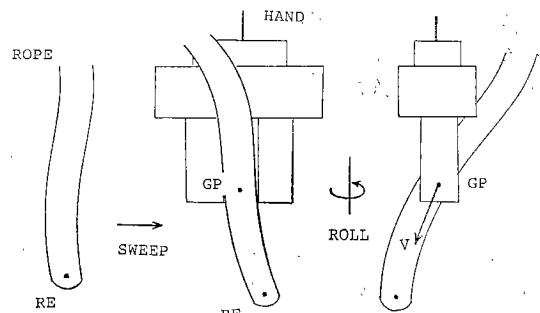


Fig. 9 Verify grasping and measure the rope-end

グの中心位置を発見するため、水平と垂直のスキャン・ラインを交互に用いる (Fig. 8 参照)。第 i 番目のスキャン・ラインにそって解析を行なうことにより、2つの紐状領域の中間位置 \mathbf{P}_i を得、その点を中心にして $i+1$ 番目のスキャン・ラインを設定する。この解析は、 $|\mathbf{P}_i - \mathbf{P}_{(i+1)}| < 1.5$ になるまで繰り返され、リングの中心位置を得る。

4.4 3次元位置計測

両眼立体視を1台のカメラで実験するために、4枚の鏡を内蔵するステレオ・アダプタを用いている。この装置により、1枚のTV画面の左側半分が、左眼入力画像に対応し、右半分が右側入力画像に対応することになる。紐の端点、リングの中心点等に対して左右の画像中から対応する点を抽出し、視覚基準座標系における位置を計算する。左右別々に求めた対応点の候補が追跡ミスによって実際の対応点でない場合のためにエピポーラ・ラインの近傍に対応点が存在するかどうかのチェックと、奥行方向の距離が予測される空間内にあるかどうかの検査を行なっている。

4.5 把握動作の確認と移動計画

Fig. 9 に紐を把んだとの操作を示す。紐の把握が成功しているかどうかを確認するために、指を閉じたあとハンドの位置を少しずらす (SWEEP)。以前存在していた場所に紐がなく、ハンドのグラスプ・ポイントの近傍で指の方向に紐が存在することを視覚により検査する。次に、紐を認識しやすいようにハンドを回転する (ROLL)。回転されたハンドの位置・姿勢情報をもとにグラスプ・ポイントから紐の端点までのベクトル \mathbf{V} を視覚により計測する。得られた \mathbf{V}_{inC} を用いて、リングへのアプローチ・ポイントの修正を行なう。ハンドの目標位置・姿勢は以下のように決まる。ハンド座標系における \mathbf{V}_{inH} は、ハンド座標系 H_{inA} を用いて、 $\mathbf{V}_{inH} = (H_{inA})^{-1} \cdot T_{ctoA} \cdot \mathbf{V}_{inC}$ となる。リング座標系 RC_{inA} に対するアプローチ点の座標系を AP_{inRC} とすると、目標ハンド座標系 H^*_{inA} をここでは、

$$H^*_{inA} = RC_{inA} \cdot AP_{inRC} \cdot \text{Trans}(-\mathbf{V}_{inH})$$

としている。ただし、 $\text{Trans}(\mathbf{V})$ はベクトル \mathbf{V} の平行移動を行なう変換行列である。

4.6 視覚による紐の誘導

紐の先端に対する円筒状の目標空間 (ガイド・チューブ) を設定する。Fig. 10 にその具体例を示す。紐の先端部がガイド・チューブ内に常に存在するようビジュアル・フィードバックをかけ、紐をリングに通す動作を実現している。ガイドチューブは RC に設定された座標系における中心座標ベクトル $C_i(C_{xt}, C_{yt}, C_{zt})$ 、長さ l_i 、半径 r_i で表現される。視覚により紐の先端がガイ

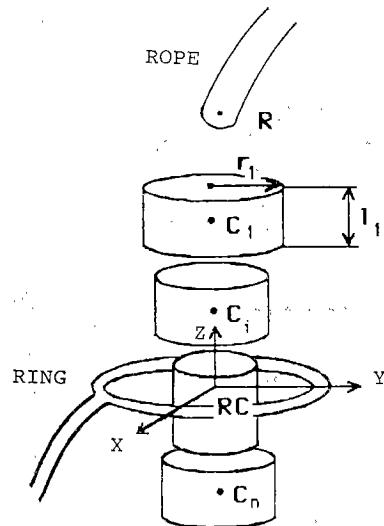


Fig. 10 Imaginary guide tube for visual feedback

ド・チューブに入っているかどうかを判定し、入っているればこのパラメタを次の誘導のために更新するという処理が繰り返される。

第 i 番目のガイド・チューブに対する判定条件を紐の先端の位置の座標系 RC における位置ベクトル $R_i = (R_{xt}, R_{yt}, R_{zt}, 1)$ を用いて次の式で与える。

$$\begin{aligned} \sqrt{(C_{xt} - R_{xt})^2 + (C_{yt} - R_{yt})^2} &< r_i \\ |C_{zt} - R_{zt}| &< l_i/2 \end{aligned}$$

紐の先端の位置ベクトル R は、視覚により計算して得られるベクトル R_{inc} と RC 座標系の表現行列 RC_{inc} を用いて $R = (RC_{inc})^{-1} \cdot R_{inc}$ として得られる。ガイド・チューブ内に紐の先端が入っていない場合には、偏差ベクトル $C_i - R_i$ より、視覚座標系における操作ベクトル $RC_{inc} \cdot (C_i - R_i)$ を行動系へフィードバックする。行動系は、与えられた操作ベクトルをハンド・アイ・キャリブレーションにより得られた変換行列によってアーム座標系にあける操作ベクトルへ変換し、ハンドの位置を修正する。修正量 x_{inA} は、

$$x_{inA} = T_{ctoA} \cdot RC_{inc} \cdot (C_i - R_i)$$

となる。

ガイド・チューブの軌道及び、サイズは、常時観察しているリングの位置をもとに修正される。フィードバック中のリング位置の計測結果を保存し、計測の失敗および異常動作を検出するために利用する。紐とリングの相対位置が異常に変化した場合にはガイド・チューブの再設定を行い、カウンタ i を減じることによって紐の誘導をやり直すことができる。

実験の写真を Fig. 11 に示す。左側の写真はロボットの視覚入力画像を写したものであり、左右両画像を表示

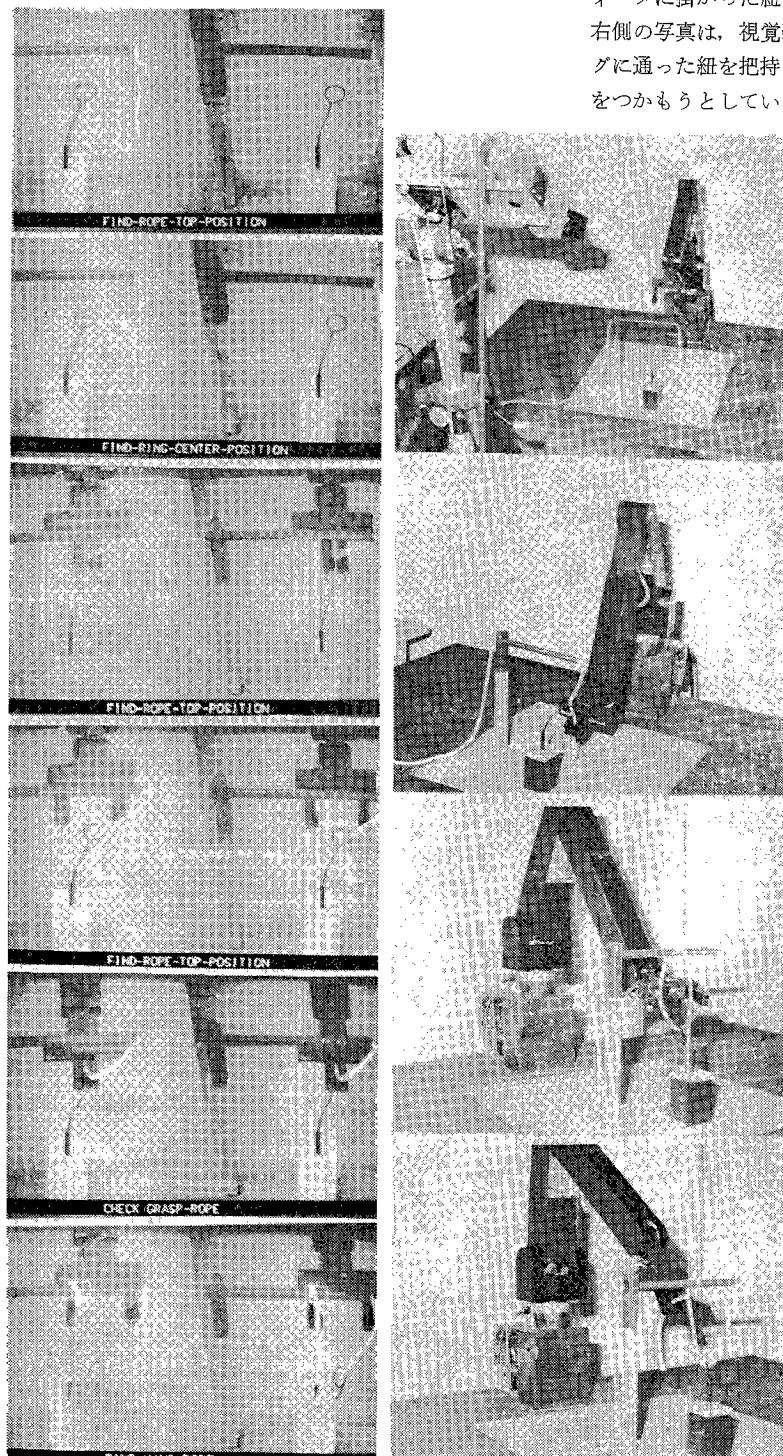


Fig. 11 Snapshots of the rope handling experiment

している。上から順に、紐の追跡、リングの探索、紐の誘導、リングに通ったことの確認、把握動作の確認、フィーダに掛かった紐の位置計測、の各場面を示している。右側の写真は、視覚装置とマニピュレータの配置、リングに通った紐を持持している場面、フィーダの下から紐をつかもうとしている瞬間、紐を結ぶために紐を引き出している場面をそれぞれ示している。

作業に要した時間は全体で約8分である。この時間の内、ミニコン上のプロセス切替とプロセス間通信、画像処理過程の表示、LISPのごみ集め(GC)にかかる時間が無視できない部分をしめている。

5. おわりに

視覚情報をを利用して動作を修正し、動作結果を基に注視する領域を制御し動作確認をするという視覚・行動系の協調動作の実現法を解析し、紐のハンドリング実験について述べた。

視覚と行動系の協調動作実験のための手法として、スキャン・ライン上の特徴抽出、両眼立体視、ハンド・アイ・キャリブレーション、ハンドの位置を利用した動作の確認、ビジュアル・フィードバック制御による対象の誘導法について述べた。

局所処理を行なう視覚機能と自動キャリブレーション機能を有するハンド・アイ・システムにより視覚・行動系間の相互のフィードバック制御を行なえば、柔軟な紐をリングに通して結ぶという操作ができるることを実験により確認した。

動作修正時の目標空間の指定方法、注視領域の制御方法、および作業内容の記述方法の一般化が今後の課題である。

謝 辞

実験に用いたシステム (COS

MOS) は昭和 58 年までに東京大学工学部機械工学科井上研究室で開発されてきたものである。COSMOS プロジェクトに貢献された諸氏に深く感謝する。特に、Lisp の処理系を作成し各サブシステムの統合を実現された電子技術総合研究所の小笠原司氏、視覚システムの基礎を作成された同所松井俊浩氏にあらためて謝意を表す。

参考文献

- 1) 石井 優, 長田 正, 永山一行, 三上芳夫, "HAND-EYEシステムの研究(I)", 電子技術総合研究所彙報, 第38巻, 第10号, pp.587-606, 1974
- 2) Y.G. Barraquand, 稲葉雅幸, 井上博允, "An Experiment with Hand Eye System", 第2回日本ロボット学会学術講演会予稿集, pp.145-148
- 3) R.O. Duda, and P.E. Hart, "Pattern Classification and Scene Analysis", John Wiley & Sons, 1973
- 4) 稲葉雅幸, 井上博允, "ハンド・アイ・システムにおける自動キャリブレーション", 第22回 SICE 学術講演会予稿集, pp.593-594, 1983
- 5) 稲葉雅幸, 井上博允, "ハンド・アイ・システムにおけるラン・タイム・キャリブレーション", 第23回 SICE 学術講演会予稿集, pp.353-354, 1984
- 6) 稲葉雅幸, 井上博允, "ハンド・アイ・システムにおける動作の監視と制御", 第22回 SICE 学術講演会予稿集, pp.595-596, 1983
- 7) 白井良明, 井上博允, "ビジュアル・フィードバックを利用したロボットの組み合わせ作業", 電子技術総合研究所彙報, 第35巻, 第3号, pp.587-606, 1974
- 8) H. Inoue, T. Ogasawara, O. Shiroshita, and O. Naito, "Design and Implementation of High level Robot Language", Proceedings of 11th International Symposium on Industrial Robots, Tokyo, 1981
- 9) 小笠原司, 井上博允, "知能ロボット・プログラミングシステム COSMOS", 日本ロボット学会誌, 第2巻, 第6号, pp.507-525, 1984
- 10) 稲葉雅幸, 井上博允, "知能ロボットプロトタイプ COSMOS の視覚システム", 第1回日本ロボット学会学術講演会予稿集, pp.31-32, 1983

稻葉雅幸 (Masayuki INABA)



昭和33年5月23日生まれ。56年3月東京大学工学部機械工学科卒業。58年3月同大学院工学系研究科情報工学専門課程修士課程修了。現在、同博士課程に在学、視覚を有するロボットの研究中。

(日本ロボット学会学生会員)

井上博允 (Hirochika INOUE)



昭和17年7月5日生まれ。40年東京大学工学部産業機械工学科卒業。45年同大学博士課程修了、工学博士。45年電子技術総合研究所入所。53年4月より同大学助教授工学部機械工学科。現在、同教授。知能ロボット総合システムの研究中。日本機械学会、情報処理学会、電子通信学会、計測自動制御学会、IEEE、ACM の会員。

(日本ロボット学会正会員)

Hand Eye Coordination in Rope Handling*

Masayuki INABA** Hirochika INOUE***

ABSTRACT

Visual information is necessary for a robot to manipulate flexible objects like ropes. Direct visual feedback and verification in three dimensional space are the keys to perform successful manipulation of flexible objects.

In this paper the important points of hand eye coordination are discussed and a rope handling experiment is described. Automatic calibration, visual verification at grasping operation, and visual feedback for guiding a object are described. The hand eye system consists of robot vision system, a general purpose manipulator, and Lisp system. Main visual functions are local image processing along a linear region and stereo vision. The experiment involves task performance such as rope-into-ring and tying a rope. Coordinating general purpose arm with stereo vision, our robot succeeds in manipulating a flexible rope.

Key words: Hand eye system, Robot vision, Stereo vision, Calibration, Visual feedback

* Received July 11, 1985 ** Tokyo Graduate School, University of Tokyo *** University of Tokyo