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——

橋 本 秀 紀* 久 保 田 孝*
佐 藤 基 夫** 原 島 文 雄*

Visual Control of Robotic Manipulator Based on Artificial Neural Network

Hideki HASHIMOTO Takashi KUBOTA
Motoo SATO Fumio HARASHIMA

This paper describes a control scheme for a robotic manipulator system which uses visual information to position and orientate the end-effector. In the scheme the position and the orientation of the target workpiece with respect to the base frame of the robot are assumed to be unknown, but the desired relative position and orientation of the end-effector to the target workpiece are given in advance. The control system directly integrates visual data into the servoing process without subdividing the process into determination of the position and orientation of the workpiece and inverse kinematic calculation. An artificial neural network system is used for determining the change in joint angles required in order to achieve the desired position and orientation. The proposed system can control the robot so that it approach the desired position and orientation from arbitrary initial ones. Simulation for the robotic manipulator with six degrees of freedom is done. The validity and the effectiveness of the proposed control scheme are verified by computer simulations.

Key Words : Robot manipulator, Visual information, Position and posture control, Neural network.

1. 緒 言

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¹⁾が盛んに行われている。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²⁾が行われている。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。

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³⁾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⁴⁾などが行われている。

本論文では、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

原稿受付 1989 年 9 月 25 日

* 東京大学生産技術研究所

** 日本無線株式会社

Fig. 1 System configuration

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。

2. 問題設定

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。この作業は次の3つの手順からなる。

- (a) 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。
- (b) 対象物をつかむ。
- (c) つかんだ対象物を他の場所に移動して置く。

ここでは(a)の手順に注目する。(a)の作業がうまく行われれば、続く(b)(c)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、Fig. 1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、Fig. 2に示すよ

Fig. 2 Transformation of coordinate system

うに4つのステップから成る座標変換が必要となる。

2.1 座標変換

＜ステップ1＞では、(1)式に示すように回転行列 R によって絶対座標系からカメラ座標系への変換を行う。ただし、絶対座標系でのカメラ位置を (X_c, Y_c, Z_c) 、オイラー角で表したカメラの姿勢を (ϕ_c, ξ_c, ψ_c) 、絶対座標系における対象物の特徴点の位置を $(^P X_i, ^P Y_i, ^P Z_i)$ 、カメラ座標系における対象物の特徴点の位置を $(^C X_i, ^C Y_i, ^C Z_i)$ とする。

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

＜ステップ2＞では、カメラ座標系における特徴点をカメラの視点から焦点距離 f だけ離れた所に設定した投影平面に透視変換を行い、次の(2)式によって表される。

$$\begin{cases} ^I X_i = f \frac{^C X_i}{^C Z_i} \\ ^I Y_i = f \frac{^C Y_i}{^C Z_i} \end{cases} \quad (2)$$

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

＜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 (ϕ, ξ, ψ) から各関節角度 (θ_j) を求めるために、逆キネマティクスを解く。

2.2 画像情報から位置・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。(1),(2)式において、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 $(^I X_i, ^I Y_i)$ ($i=1, 2, \dots, n$)であり、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 (ϕ_c, ξ_c, ψ_c) である。従って、(1),(2)式から未知パラメータを決定するためには、さらに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を求めることができる。さらに、対象物に関する位置情報から絶対座標におけるカメラの位置姿勢を求めることができる。この関係を次の式に示す。

$$\begin{cases} L_{12} = \sqrt{(^P X_1 - ^P X_2)^2 + (^P Y_1 - ^P Y_2)^2 + (^P Z_1 - ^P Z_2)^2} \\ L_{13} = \sqrt{(^P X_1 - ^P X_3)^2 + (^P Y_1 - ^P Y_3)^2 + (^P Z_1 - ^P Z_3)^2} \\ L_{14} = \sqrt{(^P X_1 - ^P X_4)^2 + (^P Y_1 - ^P Y_4)^2 + (^P Z_1 - ^P Z_4)^2} \end{cases} \quad (3)$$

しかし、実際に解を導くには、(1),(2),(3)式の非

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⁵⁾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^{6,7)}ことにとどまっており、さらにマニピュレータへの関節角度に変換する研究⁸⁾はあまりみられない。

そこで、本論文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。

3. 位置・姿勢制御法

3.1 ビジュアルコントロール

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⁹⁾。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。

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つの特徴点すべてが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。Fig. 3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。Fig. 3(a)は、現在の位置及び姿勢における対象物の画像であり、Fig. 3(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(Fig. 3(a))の特徴点4点を目標画像(Fig. 3(b))の特徴点4点に一致させるように各関節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 $\Delta\theta_j$ を求めればよい。しかし、画像上でのデータ間の差 $(\Delta X_i = X_i - X_i^*, \Delta Y_i = Y_i - Y_i^*)$ とマニピュレータの各関節への指令値 $\Delta\theta_j$ との間の関係は非線形であり、2章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。

3.2 ニューラルネットワークシステム

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¹⁰⁾を用いる。ニューラルネットワークは、入力層、隠れ層、出力層の3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。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。

3.3 学習・実行プロセス

Fig. 4 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。このとき各関節に取り付けられた内界センサによって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。**Fig. 5** 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。

4. 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。シミュレーションでは、**Fig. 6** に示すように、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。なお、シミュレータとしてNeural Works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。

4.1 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

Fig. 3 Camera images

Fig. 4 Block diagram of learning process

画像での特徴点4点の座標値を次のように設定する.

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

ここでは、特徴点4点すべてがカメラの視野内に入るように、マニピュレータの手先(カメラ)の動作範囲を、**Table 1**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 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。

4.2 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、

Fig. 5 Block diagram of execution process

Table 1 Restrictions to training patterns

θ_1	$-0.87 \sim 0.87 [\times 10^1 \text{ rad}]$	$(-5.0^\circ \sim 5.0^\circ)$
θ_2	$-5.24 \sim -4.36 [\times 10^1 \text{ rad}]$	$(-30.0^\circ \sim -25.0^\circ)$
θ_3	$-3.45 \sim 4.36 [\times 10^1 \text{ rad}]$	$(20.0^\circ \sim 25.0^\circ)$
θ_4	$-2.62 \sim 2.62 [\times 10^1 \text{ rad}]$	$(-15.0^\circ \sim 15.0^\circ)$
θ_5	$-0.87 \sim 0.87 [\times 10^1 \text{ rad}]$	$(-5.0^\circ \sim 5.0^\circ)$
θ_6	$-2.62 \sim 2.62 [\times 10^1 \text{ rad}]$	$(-15.0^\circ \sim 15.0^\circ)$

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。まず、学習パターンに対して実行プロセスを行った。そのときの特徴点4点についての画素エラーを**Table 3**に示す。**Table 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点の画素エラーを差したものである。このデータより最終エラーは-4~4画素程度に抑えられた。**Fig. 7**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。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。

4.3 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

Fig. 6 PDP network

Table 2 Training patterns

Trainig No.	Input Patterns (pixel)								Output Patterns [$\times 10^{-1}$ rad]					
	Δx_1	Δy_1	Δx_2	Δy_2	Δx_3	Δy_3	Δx_4	Δy_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	-0.87	-5.06	4.01	-2.27	-0.52	-0.87
2	18	-78	42	-55	15	14	108	-38	0.35	-4.89	3.84	-2.62	0.70	-1.57
3	28	-20	29	-2	-14	39	71	40	0.17	-5.24	4.36	-2.27	-0.35	2.44
4	-84	-146	-91	-135	-129	-98	-53	-94	-0.70	-4.54	4.19	-1.75	0.35	1.57
5	-123	-2	-116	16	-141	72	-59	40	-0.87	-4.54	3.67	-1.40	-0.52	-1.92
6	-68	-89	-57	-70	-87	-9	3	-45	-0.52	-5.24	3.84	-1.22	0.87	-2.27
7	-2	-100	-1	-87	-40	-47	39	-49	0.00	-4.54	4.01	-1.40	0.17	2.27
8	24	2	10	23	-48	46	40	80	0.00	-4.89	3.84	0.70	-0.17	2.27
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
60	-84	-140	-99	-127	-143	-100	-69	-80	-0.70	-4.36	3.84	0.17	0.70	0.87

Table 3 Image error (Learned patterns)

Error	Δx_1	Δy_1	Δx_2	Δy_2	Δx_3	Δy_3	Δx_4	Δy_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

いて適用した場合についてシミュレーションを行った。未学習データとしては、学習プロセスで学習を行った、Table 1 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。そのときの画像エラーを Table 4 に示す。また、そのときのシミュレーション結果を Fig. 8 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、なかには近づいているが、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考えられる。

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 点すべてが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。

5. 結 言

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、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

Table 4 Image Error (Not learned patterns)

Error	Δx_1	Δy_1	Δx_2	Δy_2	Δx_3	Δy_3	Δx_4	Δy_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

Fig. 7 Simulation results (Learned patterns)

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

- 1) R.L. Anderson, "Real Time Intelligent Visual Control of a Robot", IEEE Int. Workshop on Intelligent Control, pp. 89-94, 1985.
- 2) A.C. Sanderson, "Vision and Multisensor Feedback Controllers". IEEE Int. Workshop on Intelligent Control, pp. 80-81, 1985.
- 3) M. Kuperstein, "Neural Model of Adaptive Hand-Eye Coordination for Single Postures", Science, Vol. 239, pp. 1308-1311, 1989.
- 4) W.T. Milles, "Real-Time Applications of Neural Networks for Sensor-Based Control of Robots with Vision", IEEE Trans. on Systems, Man, and Cybernetics, Vol. 19, No. 4, 1989.

Fig. 8 Simulation results (Not learned patterns)

- 5) R. Tsai, T. Huang, "Uniqueness and Estimation of Three Dimensional Motion Parameters of Rigid Objects with Curved Surfaces", IEEE Trans. on PAMI-6, 1984.
- 6) 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.
- 2) 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.
- 9) 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.
- 9) 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.
- 10) D.E. Rumelhart, J.L. McClelland, the PDP Research Group, "Parallel Distributed Processing", Bradford Books, Vol. 1 & 2, MIT Press, 1986.