# 로보트의 고기능화를 위한 연구

박 혜 숙 , 장 동 식

고려대학교 산업공학과

## 요 약 문

본 연구는 로보트에 시각기능을 부여하고 새로운 제어기와 구동소프트웨어를 개발하여 로보트의 고기능화를 이룸으로써 로보트작업의 정밀도와 유연성을 향상시키는 것을 목적으로 한다.

로보트의 제어방식과 구동소프트웨어의 여러 문제점을 보완하여 새로운 제어기와 구동소프트웨어를 개발하였다. 또한 시각센서를 사용하여 로보트가 물체의 위치와 형태를 스스로 인식할 수 있도록 카메라 캘리브레이션, 이치화(Thresholding), 물체의 외곽선 추출(Contouring), 물체의 크기, 중심선 및 중심점 인지 등의 기법을 개발하여 적용하였다.

## 제 I 장 서 론

산업용 로보트는 1960년대에 처음으로 산업계에 등장하여 제품생산에 큰 몫을 담당하고 있다. 더우기 생산현장에서의 자동화 시스템이 복잡해짐에따라 단순 반복적인 작업뿐만아니라 고기능의 작업능력을 지닌 산업용 로보트가 요구되고 있다. 따라서 시각, 청각 등 정보처리기능이나 현장에 대한 계획 및 학습기능 등 인간의 우수한 동작제어능력의 일부를 로보트에게 부여함으로써 고기능화된 유연성있는 로보트를 만들려는 시도가 행해지고 있다.[5][6] 실제로 산업현장에서 로보트의 작업은 미리 교시된 고정위치에서의 pick & place 작업이 대부분이다. 따라서 작업내용이 복잡해짐에 따라 로보트 제어시간이 많이 필요하고

작업교시시간도 급증하게 되어 조작 에러는 물론 생산성측면에서도 기대에 못미치고 있다.[6]

본 연구에서는 로보트에 시각기능을 부여하여 작업자가 미리 교시하지 않아도 로보트가 대상물을 스스로 인지하고 대상물에 대한 정보를 획득하여 필요한 작업을 수행할 수 있는 소프트웨어를 개발하였다. 또한 기존 제어기의 제어방식의 문제점을 보완하여 로보트를 효과적으로 제어할 수 있는 제어기를 개발하여 고기능화된 로보트 시스템을 구축하였다.

## 제 Ⅱ장 고기능 로보트 시스템

### 제 1 절 시각 센서의 응용

#### 1.1 카메라 영상으로부터의 정보획득과정

카메라를 통해 받아들인 대상물의 영상은 계수화(Digitization)를 거쳐 컴퓨터의 메모리상에 저장된다.[1] 저장된 영상으로부터 대상물에 대한 정보를 추출하기 위하여 최적의 이미지를 만들고 이로부터 필요한 정보를 획득하게 된다. 이 과정은 크게 3 가지 단계로 나누어 지는데 첫째 영상의 이치화(Thresholding) 과정 둘째, 외곽선 추출(Contour Extraction) 세째, 대상물의 중심점(Central Point)과 방향선(Orientation)을 구하는 것이다.

##### 1.1.1 이치화(Thresholding)

이 단계는 영상이 갖는 왜곡(Distorsion)이나 잡음(Noise)을 제거하여 효과적인 영상을 얻는 과정이다. 영상은 0부터 255 까지의 그레이레벨을 갖게된다. 각 픽셀의 그레이 레벨이 두가지 값을 갖도록 제한하여, 영상의 강조와 단순화는 물론 영상의 저장, 재생과 그레이 레벨의 수학적 계산을 용이하게 하였다. 이

단계를 간단히 나타내면 다음과 같다.[2]

$$\text{IF } f(x,y) \geq T \text{ then } f'(x,y) = 1 \qquad (1.1)$$

$$\text{IF } f(x,y) < T \text{ then } f'(x,y) = 0 \qquad (1.2)$$

여기서 f(x,y)는 영상의 (x,y)좌표에서의 그레이 레벨을 나타내고 T는 영상을 이치화 시키는 역치값(Threshold Value)이다. f'(x,y)는 이치화가 된 후의 (x,y) 좌표의 그레이 레벨이다. 이 과정이 끝나면 영상의 왜곡과 잡음은 제거되고 대상물은 배경과 구별되어 강조된다.[5]

1.1.2 외곽선 추출 (Contour Extraction)

이 과정에서는 이치화 과정이 끝난 후, 대상물에 관한 구체적 정보를 구하는 첫단계로서 대상물의 외곽선을 추출하여 대상물의 크기나 모양 등에 관한 정보를 얻는 것이다. 여기서 외곽선이란 대상물과 배경(Background)의 불연속성이 존재하는 경계면에서의 픽셀들의 집합을 뜻한다.[5] 본 논문에서 사용한 외곽선 추출방법은 Contour Following 알고리즘이다. 이것은 대상물 둘레에 있는 임의의 한점을 출발점으로하여 본래 그레이 레벨 영상에서 단계적으로 대상물의 외곽선을 찾아가는 방법이다. 일단 대상물 주위의 임의의 한점을 시작점으로 외곽선 경로를 따라 해당 픽셀의 뒤에 오는 픽셀을 평가하여 시계 방향으로 대상물을 추적하게 된다. 이 알고리즘이 간단하고 빠르며 노이즈에 영향을 덜 받는다는 장점때문에 본 논문에 이용하였다.

1.1.3 대상물의 중심점(Central Point)과 방향선(Orientation) 구하기

대상물의 위치와 모양은 중심점과 방향선을 구함으로써 알 수 있다. 픽셀들의 1차모멘트와 2차모멘트를 구하여 이로부터 중심점과 방향선을 구했다.[11] 식(1.3)부터 식(1.8)은 중심점과 방향선을 구하는 식이다. 여기서 M(m,n)은 픽셀의 m-n차 모멘트이고 ( $\overline{x_k}$ , $\overline{y_j}$ )는 픽셀의 중심점을 나타내며 U(m,n)는

centroid 에 n차 모멘트이다. 그리고 행렬 E는 U의 eigenvector들을 나타낸다. 식 (1.5)와 (1.8)로부터 대상물의 중심선과 방향선, $\theta$를 구한다.

$$M(m,n) = \frac{1}{J^n K^m} \sum_{j=1}^{J} \sum_{k=1}^{K} (x_k)^m (y_j)^n F(j,k) \tag{1.3}$$

$$M(0,1) = \frac{1}{J} \sum_{j=1}^{J} \sum_{k=1}^{K} y_j F(j,k) \qquad M(1,0) = \frac{1}{K} \sum_{j=1}^{J} \sum_{k=1}^{K} x_k F(j,k) \tag{1.4}$$

$$\bar{x}_k = \frac{M(1,0)}{M(0,0)} \qquad \bar{y}_j = \frac{M(0,1)}{M(0,0)} \tag{1.5}$$

$$U(m,n) = \frac{1}{J^n K^m} \sum_{j=1}^{J} \sum_{k=1}^{K} [x_k - \bar{x}_k]^m [y_j - \bar{y}_j]^n F(j,k) \tag{1.6}$$

$$\mathbf{U} = \begin{bmatrix} U(2,0) & U(1,1) \\ U(1,1) & U(0,2) \end{bmatrix} \qquad \mathbf{E} = \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix} \qquad \mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

$$\mathbf{E}^T \mathbf{U} \mathbf{E} = \mathbf{\Lambda} \tag{1.7}$$

Let $\lambda_M = \mathrm{MAX}[\lambda_1, \lambda_2]$

$$\theta = \tan^{-1}\left\{\frac{\lambda_M - U(0,2)}{U(1,1)}\right\} \tag{1.8}$$

1.2 카메라 캘리브레이션 (Camera Calibration)

본 연구에서는 카메라와 로보트가 분리된 작업 환경에 놓여있다. 그러므로 두 장치는 <그림 1>에서와 같이 서로 다른 좌표계(Coordinates)를 갖는다. 이로 인하여 카메라좌표계(Camera Coordinate)상에서 획득한 대상물에 관한 정보들은 로보트를 제어하는데 그대로 이용될 수가 없게 된다. 따라서 카메라좌표계에서 획득한 정보를 로보트좌표계의 정보로 변환시켜 주어야 한다. 이를 위해 카메라 캘리브레이션이 필요하다. 본 연구에서 이용한 카메라 캘리브레이션 알고리즘은 카메라좌표계와 로보트좌표계 사이의 관계를 4 x 4 변환행렬로 나타내고 이 행렬을 이용하여 카메라좌표계상에서 본 물체의 위치정보로부터 로보트좌표계상의 위치를 찾아내는 것이다. 이 알고리즘에서 두 좌표계사이의 관계는 다음과 같이 표현된다.

$$C = A * W$$

$$W = ( W_x, W_y, W_z, 1 ) , C = ( C_x, C_y, C_z, 1 )$$

여기서 W는 로보트좌표계에서의 임의의 한점이고 C는 카메라좌표계에서의 W에 대응하는 점이다. 그리고 A 행렬은 4 x 4 변환행렬(Transformation Matrix)이다. 그런데 현재의 작업환경 하에서 카메라 한대를 사용하였기 때문에 3차원 공간에서의 임의의 한점은 카메라의 이미지 평면상의 점으로 대응되므로 물체의 높이에 대한 정보는 무시되게 된다.[13][8] 실제로 카메라 캘리브레이션 문제는 A 행렬의 역행렬을 구하고 카메라좌표계상에서 본 물체의 위치와 곱함으로써 로보트좌표계상의 대응위치를 구하는 것이라 볼 수 있다. 이를 행렬식으로 다음과 같이 나타낸다.

$$C' = A' * W' \tag{1.3}$$

$$(A')^{-1} = W' * ( C')^{-1} \tag{1.4}$$

$$W' = (A')^{-1} * C' \qquad (1.5)$$

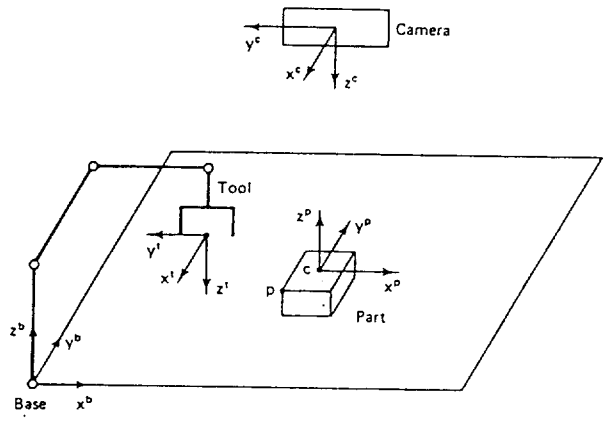
위 식에서 C'은 3 x 1 행렬, A'은 3 x 3 행렬이고 W'은 3 x 1 행렬이다.

본 논문에서는 카메라좌표계상의 기지의 3점과 이에 대응하는 로보트좌표계상의 3점을 구하여 A'행렬을 구했고 식(1.5)으로부터 카메라좌표계에서 얻은 대상물의 중심점을 로보트좌표계상의 중심점으로 변환시켰다. 변환된 대상물의 중심점은 로보트제어기로 전달된다.

## 제 2절. 로보트 제어 부분

### 2.1 SCORBOT ER - Ⅲ의 특징

본 연구에서 이용한 로보트는 SCORBOT ER-Ⅲ로서 로보트 본체와 이를 제어하는 로보트제어기(Controller)로 나누어져 있다. 컴퓨터로 로보트를 제어하기위해서는 제작사에서 제공하는 램상주 구동소프트웨어를 호출하여 사용한다.[4] 컴퓨터와 로보트제어기 사이의 통신방식은 RS-232C를 통한 비동기 통신방식이다.




< 그림 1 > 서로 다른 좌표계상에 놓여있는 카메라, 로보트와 대상물체

## 2.2 SCORBOT-ERⅢ 제어의 특징

### 2.2.1 모터 구동시스템(Drive system)

로보트의 모든 조인트는 DC 서버 모터에 의해 구동된다. 각 모터는 closed-loop로 제어됨으로써 움직인 각도와 방향이 feedback된다. 이를 위하여 각 모터마다 rotary optical encoder가 장착되어 있다. 모터 구동시스템은 모터와 조인트사이가 gear와 belt로 연결되어 있는 "Indirect drive system"이다. 또한 모터들의 transmission ratio에 의해 각 축의 회전율을 구할 수 있는데 base는 0.094, shoulder, elbow는 0.1175 이고 pitch, roll은 0.458의 회전율을 갖는다.

### 2.2.2 Controller Operating System Commands

호스트 컴퓨터에서 보내는 모든 명령은 아스키 코드로 전환되어 로보트 제어기로 순차적으로 전달된다. 명령은 세가지 타입으로 나누어진다.

(1) Uni-directional Commands : 구동명령

(2) Regular bi-directional Commands : Status Check 명령

(3) Special bi-directional Commands : 모터의 회전각 측정명령

위와같은 제어방식에는 몇가지 문제점이 있다. 각 축의 독립적인 제어는 가능하지만 동시제어가 불가능하고 속도가 느리며 로보트의 절대위치 파악이 어렵다.

### 2.2.2 스코보트 구동소프트웨어

스코보트 구동소프트웨어로 Utility+ software가 있다. 이 소프트웨어는 램상주 프로그램으로서 Assembley언어, basic, C언어와 인터페이스가 가능하다.[12] 따라서 사용자는 이 소프트웨어에서 제공되는 함수(function)를 이용

하여 프로그래밍을 할 수 있다. 그런데 이 소프프웨어를 이용하여 로보트를 제어할때는 관절값을 미리 입력시켜야한다. 제어기와 소프트웨어의 이와같은 문제점때문에 로보트는 저수준의 작업능력을 갖게되고 유연성이 떨어져 다양한 작업을 수행하지 못하는 경우가 많다.

2.3 개선된 제어기의 개발

2.2절에서 지적한 문제점을 해결하여 로보트의 유연성을 높이고 다양한 기능을 수행할 수 있도록 고기능화된 새로운 로보트제어기와 구동소프트웨어를 개발하였다.

2.3.1 개선된 제어기의 특징

1) 정밀도의 향상

2) 각 축의 독립적인 속도조절 기능과 속도의 향상

3) 각 축의 동시제어와 독립제어

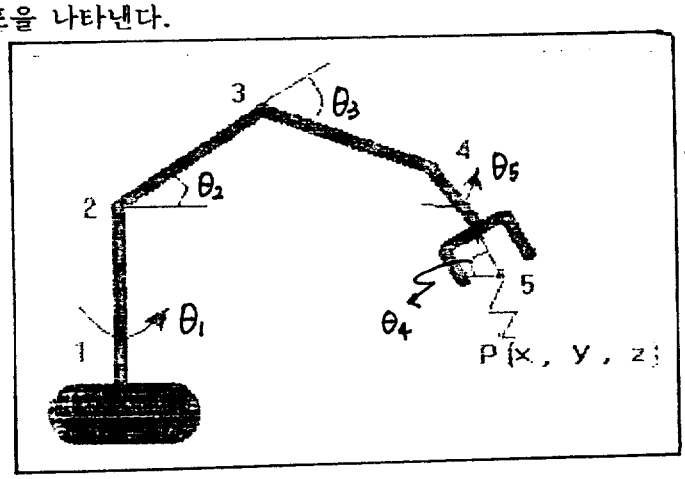
4) 각 축의 현재위치에 대한 정보 획득이 가능

2.3.2 개선된 구동 소프트웨어

기존의 제어기로는 대상물의 위치를 미리 기억시켜 고정된 위치에서만 대상물에 대해 작업할 수 있거나 시행착오적인 방법으로 로보트를 제어했었다. 하지만 본 연구에서는 로보트의 제원을 토대로 대상물의 좌표만 입력하면 각 축의 관절값을 구하여 각 축을 직접 제어할 수 있는 소프트웨어를 개발하였다. 본 논문에서는 기하학적인 해석에 의해, 목표점이 입력되면 로보트의 상호 인접한 연결축 사이의 관계에 의하여 그리퍼(Gripper)의 위치로부터 각 관절값인 $\theta$값을 구해내는 방법을 이용하였다.[3][9]

< GEOMETRIC INVERSE KINEMATICS >

<그림 2>는 로보트 각 축의 관절값을 구하기 위해 사용되는 변수들 및 각 관절을 좌표계상에서 나타낸 것이다. 여기서 P(x,y,z)는 공간상에서 로보트의 그리퍼가 놓여야할 좌표을 나타낸다.




< 그림 2 > 좌표계상에서의 각 관절위치

(1) $\theta_1$

$\theta_1$는 허리관절의 회전각을 나타낸다.

$$\theta_1 = \tan^{-1}(y/z) \qquad (2.1)$$

(2) $\theta_2$

$\theta_2$는 수평면과 어깨관절(shoulder)사에에 이루는 각을 나타낸다.

$$\theta_{21} = \cos^{-1}(len23^2 + len24^2 - len34^2 / (2len23len24)) \qquad (2.2)$$

$$\theta_{22} = \cos^{-1}(len24^2 + len12^2 - len14^2 / (2len24len12))$$

$$\theta_2 = \theta_{21} + \theta_{22} - \Pi/2 \qquad (2.3)$$

(3) $\theta_3$

$\theta_3$는 어깨관절(shoulder)과 손목관절(elbow)이 이루는 각을 나타낸다.

$$\theta_3 = \cos^{-1}(len24^2 + len23^2 - len34^2 / (2len23len24)) \qquad (2.4)$$

(4) $\theta_4$

$$\theta_4 = \cos^{-1}( len45^2 + len56^2 - len46^2 / (2len45len56)) \qquad (2.5)$$

## 제 3 절 실험 및 결과

### 3.1 실험환경

#### 3.1.1 구성장치

컴퓨터는 Coprocessor가 장착된 IBM-PC 386을 사용했고 카메라는 HITACHI사의 캠코더를 사용하였다. 그리고 Frame grabber는 영상의 획득과 저장, 재생,처리를 위한 것으로서 ITI사의 PCVISIONplus Model을 사용했다.[7] 로보트는 ESHED ROBOTEC사의 SCORBOT-ER Ⅲ 연구용 로보트이다.

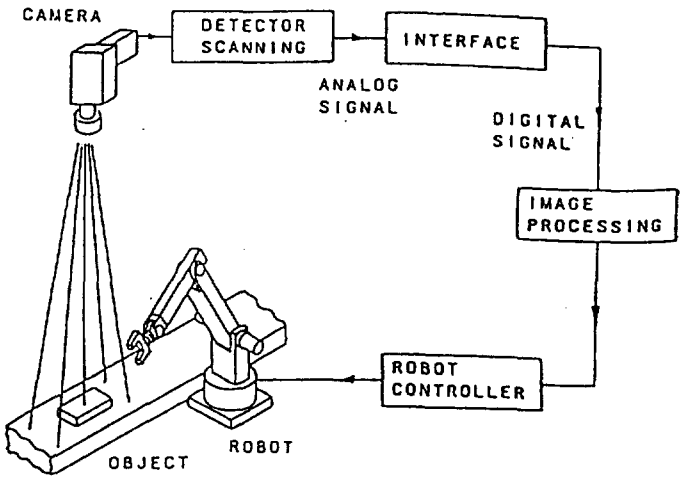
#### 3.1.2 실험내용

작업대 위에 놓인 대상물의 영상을 카메라를 통해 받아들인 후, 영상처리를 통해 필요한 정보를 획득한다. 이 정보는 카메라 캘리브레이션을 거쳐 로보트 제어기쪽으로 전달되고 제어기는 Inverse Kinematics에 의해 관절값을 구하여 로보트를 구동시킨다. 기존의 제어기와 새로 개발된 제어기에 대하여 위의 과정을 각각 적용함으로써 두 제어기 사이의 성능을 분석해 본다.성능의 척도로는 모터 제어의 정밀도, 로보트의 작업속도와 작업오차를 사용한다. 본 시스템의 구성은 < 그림 3 >과 같다.

### 3.2 실험결과

새로운 제어기는 15V 전압을 0부터 FF(255)로 쵸핑하여 전원전압을 가변시킴으로써 속도조절이 가능해졌고 속도가 크게 빨라졌으며 정밀도는 약 4배 향상되었다. 또한 디지탈 신호를 빠른 시간내에 처리할 수 있는 320C25 CPU를 채용하여 샘플링 타임을 2 usec로 줄였고 각 샘플링 타임마다 스피드에 관한 데이타를 유지하면서 다른 모터에 대한 콘트롤을 할 수 있게되어 각 축을 동시 제어할 수 있게 되었다. 그리고 영상정보를 이용하여 로보트의 작업수행오차는 기존 제어

기에서는 평균 5 ㎜였으나 새로운 제어기에서는 평균 1 ㎜로 줄었다.




< 그림 3 > 로보트 비전 시스템

## 제 Ⅲ 장 결 론

본 연구에서는 로보트에 시각기능을 부여하여 로보트가 대상물을 스스로 인지한다. 로보트는 대상물로부터 필요한 정보를 획득함으로써 작업자가 미리 입력시킨 교시없이도 작업을 정확히 수행할 수 있다. 로보트 구동소프트웨어와 기존 제어기가 가진 제어방식의 문제점을 보완하여 보다 효과적으로 로보트를 제어할 수 있는 제어기를 개발하였다. 새로 개발한 제어기의 정밀도는 기존의 제어기보다 4배 향상되었고 속도조절과 동시제어가 가능해졌으며, 로보트의 작업수행 오차도 감소되었다. 본 논문에서는 대상물의 중심점, 오리엔테이션, 크기에 관한 정보를 구했는데 대상물에 대한 패턴인식(patten recognition)기능을 갖추고 신경회로망이나 AI 기법을 적용하면 더욱 유연성있는 고기능 로보트 시스템을 구축할 수 있을 것이다.

## 참고문헌

[1] Ballard, D. H. and Brown, C. M., "Computer Vision", Prentice Hall Inc.New Jersey, 1982.

[2] Fairhurst, M. G.,"Computer Vision for Robotics System :An introduction", Prentice Hall, New Jersey, 1988.

[3] J. J. Craig, "Introduction to ROBOTICS Mechanics &Control", ADISSON WESLEY PIBLISHING COMPANY, 1955.

[4] HAIM SCHLEIFER B.Sc.,Eng."SCORBOT EDUCATIONAL ROBOT TEXTBOOK 2", 1984.

[5] Mansour R. and Waldemar K., "A Research Paradigm in Human-Robot Interaction",International Journal of Industrial Eegonomics, Elsevier, pp.59-71, 1990.

[6] Nof, S. Y., "Robot Ergonomics: Optimizing robot work", Handbook of industrial robot, pp.549-604, 1985.

[7] PCVISIONplus FRAME GRABBER USER'S MANUAL, ITI, 1987.

[8] R. C. Gonzalez and P. Wintz, "Digital Image Processing", 2nd Eds Adisson Wesley, 1987.

[9] Schalkoff, R. j., "Digital Image Processing and Computer Vision", John Wiley & Sons, 1989.

[10] SCORBOT ER-Ⅲ USERS' MAMUAL, ESHED ROBOTEC LIMITED, 1988.

[11]William K. Pratt," Digital Image Processing", 2nd Eds,John Wiley & Sons inc, 1991.

[12] SCORBOT ER-Ⅲ UTILIT+ MAMUAL,ESHED ROBOTEC LIMITED, 1988.

[13] ROBERT J. SCHILLING,"FUNDAMENTALS OF ROBOTICS Analysis and Cntrol", Prentice - Hall International Editions, 1991.