

조립용 로봇의 비전 시스템에 관한 연구
A Study on a Vision System for Assembly Robots

*최봉열, **김경진, **오상득

** 변 중 남

* 경북대학교 공과대학
전자공학과

** 한국과학기술원
전기및 전자공학과

1. 서 론

1960년대 부터 등장한 산업용 로봇은 오늘날 산업 현장에서 조립, 도장, 검사, 용접등 인간이 할 수 없거나 하기 힘든일 또는 단순하고 반복적인 작업에 널리 응용되고 있는 바 최근 반도체 기술의 발달및 저렴하고 우수한 마이크로 컴퓨터의 발달과 더불어 단순 반복적인 작업에 대항하는 생산수단으로서 뿐만 아니라 비전 센서등의 여러가지 센서를 로봇트에 적용하여 고도의 인공지능을 갖춘 로봇트및 더 나아가 무인화 생산공장으로 까지 발전하게 되었다.

본 논문에서는 여러 작업공간을 가진 로봇트 시스템을 위한 비전 시스템에 관하여 다루었는데 그 응용으로서 로봇트 팔 3대와 카메라 비전센서 3대, 컨베이어 벨트로 이루어진 소규모 무인화 공정 시스템의 개발에 적용시켰다.

사용자가 원하는 작업 공간내의 영상 데이터를 받기 위하여 세대의 카메라중에서 원하는 작업공간에 사용된 카메라를 선택하여 사용자가 볼 수 있도록 주컴퓨터에서 통제 가능한 비전 데이타 멀티프렉스 (Vision data Mux : VDM)회로를 개발하였다.

본 논문에서는 Binary 영상해석을 이용한 로봇트 비전 시스템을 연구한 바 16 비트 마이크로 컴퓨터인 Z8000DM을 전담 프로세서로서 사용하였다. 화면내의 여러 물체를 찾아내고 임의의 기하학적인 모양의 물체에 대한 상세한 묘사를 위해 Boundary-Tracking 알고리즘을 사용하였는데 처리속도를 빨리 하기 위해 경계를 찾아 나가면서 Tracking을 동시에 수행하는 알고리즘을 개발하였다.

또 주 컴퓨터에서 처리가 끝난 영상 데이터를 사용자가 볼 수 있도록 하기위해 graphic display controller (GDC)를 설계,제작하였다.

2. 비전 시스템의 구성

비전 시스템은 Visual sensing을 위한 3대의 카메라, video digitizer, vision data Multiplexer(VDM), 영상처리를 위한 Single board computer, 처리결과를 나타내는 CRT와 이를 제어하는 graphic display controller로 구성된다. 로봇트 시스템의 주 컴퓨터로는 PDP 11/34를 사용하였다. 센서로는 가볍고 소형인 CCD 카메라 2대와 CCTV 카메라 1대를 사용하였고 VDM은 설계, 제작하였다.

Video digitizer는 [5]에서 제작한 것을 사용하였고 주 컴퓨터에서 보내온 명령을 해석하고 그에 따른 영상 해석을 맡은 Single board computer로는 Zilog 회사의 Z8000 development module 를 사용하였다.

Digitizer에서 얻어진 영상은 apple용 Monitor에 나타내었고 주 컴퓨터에서 처리된 데이터를 보기 위해서는 SHARP회사의 CRT 를 사용하였다. 이를 제어하기 위해 GDC를 설계,제작하였다. 본 로봇트 시스템의 hardware block diagram을 (그림 2)에 도시하였다. 사용자는 Terminal을 통해 명령을 내리면 이는 주 컴퓨터인 PDP 11/34에 의해 해석되어 serial port 를 통해 Z8000DM으로 Command Code가 보내지게 되고 Z8000DM는 받아들인 명령에 의해 serial line을 통해 VDM 회로에 신호를 보

내 3대의 카메라중 원하는 1대의 카메라를 선택하게 된다. 선택된 카메라로 부터 영상 데이터를 받아들여 Digitizer 에서 이를 digitize 시킨 후 parallel port를 통해 Z8000DM 으로 전송된다.

Z8000DM은 영상데이터를 처리하여 그 결과를 parallel port 를 통해 주컴퓨터로 보내고 주컴퓨터에서 처리가 끝난 데이터는 serial port를 통해 GDC에 적합한 명령어로 Coding된 후 전송된다. GDC는 받은 명령을 해석하여 RGB 분리형 CRT에 video signal을 보내 Display 시킨다.

상기에서 설명한 비전 시스템을 제어하기 위해 Z8000DM에서 주컴퓨터와의 통신및 명령어 해석을 하는 프로그램과 binary 영상해석 program, 주컴퓨터인 PDP 11/34에서 Z8000DM 과의 통신및 제어를 위한 프로그램과 GDC를 제어하는 프로그램, GDC의 Initialization 프로그램, 주컴퓨터로 부터온 데이터 해석및 처리, 주컴퓨터와의 통신을 하는 프로그램을 개발하였다.

3. Binary 영상을 얻는 방법

사용자는 software를 통해 VDM 회로를 동작시켜, 세대의 카메라중 원하는 카메라를 선택할 수가 있다 [3.1]. 여기서 VDM 회로의 구성은 그림과 같이 설계하였다. VDM 회로를 통해 카메라로 부터 들어온 video signal은 video digitizer를 통해서 digitizer된 영상을 얻게 된다.

binary 영상은 gray-scale 영상이나 연속적인 명암을 가진 영상을 단지 흑백 두가지 색깔을 가지게 하는 thresholding process 에 의해 얻어지게 된다. 각 pixel의 밝기를 common threshold level과 비교하여 level을 넘으면 '1'(백)의 값을 그외는 '0'(흑)의 값을 지정하는 global thresholding approach를 채택하였다. Z8000DM에서 프로그램에 의해 video digitizer내 D/A converter의 출력전압을 변화시켜 threshold level을 조정하였으므로 속도가 빠른 장점이 있다.

4. Blob을 찾는 방법

하나의 영상속에 여러개의 blob이 존재할 경우 각각의

blob에 대한 특징을 추출하기 위해서는 먼저 각 blob을 찾아야 한다. 새로운 blob을 찾는 방법은 raster scan을 이용하였다. 새로운 blob이 인식되면 raster scan을 중지하고 인식된 새로운 blob의 완벽한 description을 위해 Boundary tracking 알고리즘을 이용한다. Boundary tracking이 끝나면 또 다른 blob을 찾기 위해 raster scan은 다시 재개된다.

영상안의 다른 blob을 찾기위해 raster scan을 계속할때 이미 찾아진 blob을 여러번 만나게 되는데 이를 다시 새로운 blob으로 인식되지 않는 방법을 고쳐야 한다. 그래서 boundary tracking에 의해 찾아진 blob의 경계점을 저장하는 memory plane을 따로 잡아 이를 해결하는 방법을 개발하였다. [7]에서는 먼저 영상을 Boundary extraction을 행하여 경계만을 찾은후 경계를 tracking 하는 방법인데 비해 본 논문은 이 두가지 동작을 동시에 수행하는 방법을 사용한다는 단점을 가지고 있지만 bit operation을 줄여 속도를 증가시켜 주었다. blob의 경계를 찾는 알고리즘에 대한 흐름도는 fig.[4.1]에 있다.

5. Boundary tracking 알고리즘

영상 평면속의 blob중 찾아진 새로운 blob의 특징을 추출하기 위해 boundary tracking 알고리즘을 사용하였다. blob의 경계는 8-Directional chain-coding scheme과 4-Directional chain-coding scheme으로 표현할 수 있다. 두 scheme의 방향들은 그림 [4.2]에 나타내었고 그림 [4.3]에는 같은 blob에 대한 두 scheme의 경계를 나타내었다. 지난 point에 대한 각 경계점의 방향은 지난 point에 대한 vector로서 나타낼 수 있는데 이것은 2차원적인 영상의 정보를 가진 1차원적인 묘사로서 이루어지는데 이런 방향의 chain-code는 Freeman [1]에 의해 폭넓게 연구되어져 있다.

본 논문에서는 8-Dirn 방법을 채택하여 실험하였다. boundary tracking 알고리즘은 시계 반대방향으로 하여 처음 시작한 경계점이 나올때 까지 연속적으로 연결된 경계점을 찾아 나가는 방법이다.

다음의 경계점을 찾기 위하여서는 지금의 경계점을 중심적으로 하여 3 x 3 window를 사용하여 인접한 8방향의 pixel들을 시계반대 방향으로 조사해 나가는데 이때 '0'에서 '1'로 transition될 때 그 '1'이 다음 경계점이 되고 이때의 방향이 vector 방향이 된다. 위와같이 다음 경계점을 찾기 위해 8방향의 pixel을 조사해 나가는 것을 local-tracking이라 칭한다. local tracking을 함에 있어 조사를 시작하는 점이 있는데 이 방향을 출발 방향점이라 칭한다. 경계점에서의 local tracking 때는 지난 local tracking때 이미 조사를 받은 pixel이 존재하기 때문에 인접한 8방향의 pixel을 모두 조사할 필요가 없다. 표 4.1에 전 vector 방향에 대한 출발 방향점에 대해 나타내었다. 그러므로 직선인 부분에서는 단지 한두개의 pixel만을 조사해도 local-tracking이 가능해진다.

Last vector Direction	Starting Direction
0	7
1	7
2	1
3	1
4	3
5	3
6	5
7	5

(표 4.1)

6. 특징추출

Binary 영상으로 부터 blob의 면적, 중심, 둘레, Orientation 등과 같은 기하학적인 특징들은 boundary-tracking 알고리즘으로 추출되어 진다. blob의 수는 영상내에 존재하는 blob중 면적이 정해진 threshold 값보다 큰 것의 수를 나타낸다. 그보다 작은 면적을 가진 blob은 noise로 간주되었다. blob의 면적은 각 vector와 임의의 선 (여기서는 영상 좌표에서 y=0인 선) 사이의 면적의 합으로서 나타낸다. 각 vector에 대한 면적의 증가분을 표 4.2에 나타내었다.

vector Direction	Δ Area	2Δ Area
0	0	0
1	$Y + 1/2$	$2Y + 1$
2	Y	$2Y$
3	$Y - 1/2$	$2Y - 1$
4	0	0
5	$(Y-1/2)(-1)$	$-2Y + 1$
6	$-Y$	$-2Y$
7	$(Y+1/2)(-1)$	$-2Y - 1$

(표 4.2)

$$\text{Area} = \sum \Delta \text{Area} = 1/2 \sum 2\Delta \text{Area}$$

중심점은

$$x = \frac{1}{n} \sum_1^n x \quad y = \frac{1}{n} \sum_1^n y \quad \text{where } x, y = \text{중심점}$$

$n = \text{경계점의 수}$

7. Command Mode

Z8000DM 은 주컴퓨터로 명령을 받으면 그에 해당하는 일을 수행한 다음, 그에 필요한 데이터를 Parallel port를 통해 주컴퓨터로 넘겨준다. Z8000DM이 command를 받고 해석하는 flow chart를(그림 4.5)에 나타내었다. 각 command에서의 하는 일을 표 4.3에 보았다.

COMMAND	CODE	DESCRIPTION
CAM1	11	CAMERA1 Set
CAM2	12	CAMERA2 Set
CAM3	13	CAMERA3 Set
Blobn	21	blob의 수, 중심좌표
AREA	31	면적과 중심좌표
BOUND	32	중심좌표의 경계좌표

(표 4.3)

8. Display System

Display 시스템은 digitize된 영상을 나타내는 Monitor와 주컴퓨터에서 처리가 끝난 데이터를 hard copy하는 XYPLOTTER와 CRT 화면으로 볼 수 있게한 GDC 로 구성되어 있는데 여기서 GDC에 대해서만 언급하였다. GDC(Graphic Display Controller)의 hardware description을 살펴보면 Z80A를 CPU로 하고 UART로는 Z80A-SIO/0를 사용하고 graphic display용 IC로는

μPD7220을 사용하였으며 CLOCK Cycle은 2.4576MHz로 하였다. 시스템 블록도는(그림 5)에 나타내었다. Software는 크게 주컴퓨터에서 GDC를 제어하는 프로그램과 GDC에서 주컴퓨터에서 보내온 데이터를 해석하고 처리하는 프로그램으로 나눌 수 있다.

주컴퓨터내의 프로그램들은 LINE, PU(pen-up), PD(pen-down), CHAR(N, Character), CLEAR(N, Character), CLEAR와 같은 subprogram으로 구성되어 있어 사용자가 쉽게 사용할 수 있도록 하였다. GDC내의 프로그램은 Initialization, 주컴퓨터와의 통신, 데이터 처리를 수행하는 프로그램으로 구성된다.

7. 실험 결과

Global thresholding 방법을 사용하여 그림 6.1에 나타난 화면으로 부터 binary 영상을 얻는 결과가 그림 6.2에 나타내었다. 여기서 화상의 resolution은 128 x 128로 하였다. 그림 6.2에 보인바와 같이 level 이 18(HEX)일때 적당한 영상을 얻을 수 있으므로 그 값으로 고정시켜 실험하였다. 여러 blob이 존재하는 binary 영상에서 blob을 찾는 방법에 대한 결과는 그림 6.4에 나타내었다. boundary-tracking 알고리즘에 대한 결과로는 그림 6.4에 나타내었다. 각 처리과정의 소요시간을 살펴보면,

sampling time : 30 msec
 영상 해석 : 30-70 msec
 HOST로 32bytes Data transfer : 85 msec
 HOST로 4bytes Data transfer : 15 msec

각 command별 처리 속도

Blobn(21) : 160-190 msec
 AREA(31) : 80-110 msec
 BOUND(32) : 400-430 msec

8. 결 론

본 논문에서는 visual measurement, 영상해석, Display 등을 처리하는 실시간 비전 시스템을 개발하였다. binary 영상은 global thresholding에 의해 구하였고

영상해석으로는 Boundary map을 이용하여 Boundary extraction과 tracking을 동시에 수행하여 시간을 단축시키는 알고리즘을 개발하였다. 격리된 작업공간의 상호연관 작업을 위해 3대의 카메라를 사용한 바 이를 제어하기 위해 VDM 회로를 설계, 제작하였고 주컴퓨터에서 처리가 끝난 데이터를 나타내기 위해 raster 주사형 CRT에 문자나 도형을 나타낼 수 있는 GDC를 설계, 제작하였다. 16 비트 마이크로 컴퓨터로 영상해석을 함으로서 저렴한 implementation 할 수 있고 실시간 제어를 가능케 하였다.

본 논문에서 개발한 실시간 비전 시스템은 시각 로봇의 그림 짜맞추기에 관한 공동 연구과제에 적용되었다[7].

참 고 문 헌

1. Freeman, H. "Computer Processing of Line-Drawing Images", Computing Surveys, Vol.6, No.1, March 1974, pp.57-97.
2. J.S. Weseka, "A Survey of Threshold Selection Techniques", CGIP.Vol. 5, 1976, pp. 382-399.
3. UPD7220 Manual, NEC
4. 원종화, 오상목, 운명중, 변중남, "로봇 암두대를 이용한 조립용 로봇 시스템의 설계에 관한 연구(2)-매니플레이터 제어기", 대한전자공학회, 하계종합학술대회 논문집, Vol 7, No. 1, 1984
5. 김경진, 변중남, "동적 물체 포착을 위한 로봇 비전 시스템" 대한전자공학회 합동학술발표회 논문집, Vol.8 No. 3, 1984
6. 오상목, 김경진, 이지홍, 조영조, 유수업, 최봉열, 변중남, "조합기능을 갖는 지능 로봇 시스템에 관한 연구", 대한전기학회 계속제어 연구회 추계학술대회 논문집, 1984
7. P.W. Kitchin, "Processing of binary images"

8. 원종화, 오상록, 윤명중, 변중남, "로봇트 암 두대를 이용한 조립용 로봇트 시스템의 설계에 관한 연구(2) -매니플레이터 제어기", 대한전자공학회, 학계종합학술대회 논문집, Vol. 7, No. 1, 1984
9. 최봉열, "조립용 로봇트의 비전 시스템에 관한 연구", 한국과학기술원, 석사학위 논문, 1980

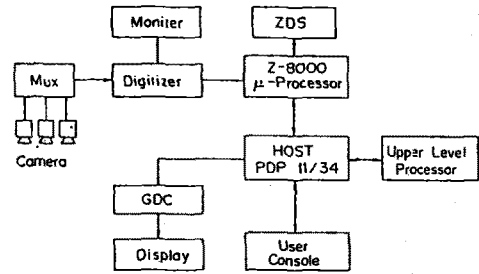


그림 2. 비전시스템 블록도

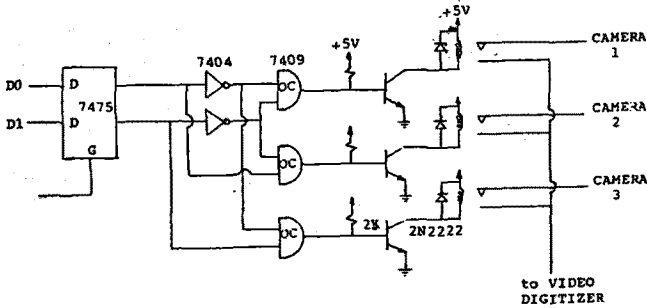


Fig. 3.1 VDH Circuit

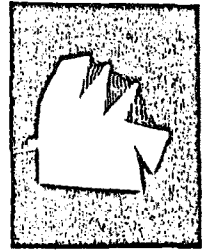
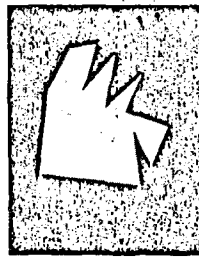
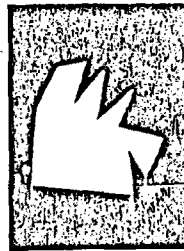


Fig. 6.1 real image

a) level=10(HEX)
Fig. 6.2 digitized image



b) level=18(HEX)

c) level=30(HEX)

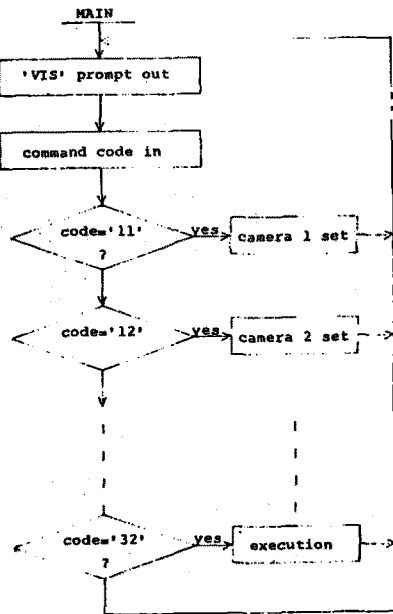


Fig. 4.5 Command Analysis Flow Chart

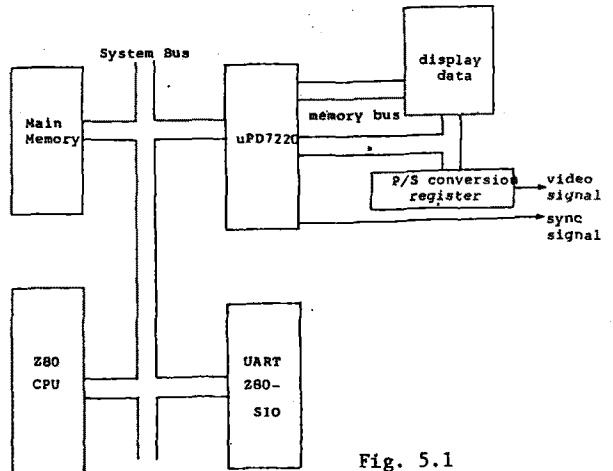


Fig. 5.1

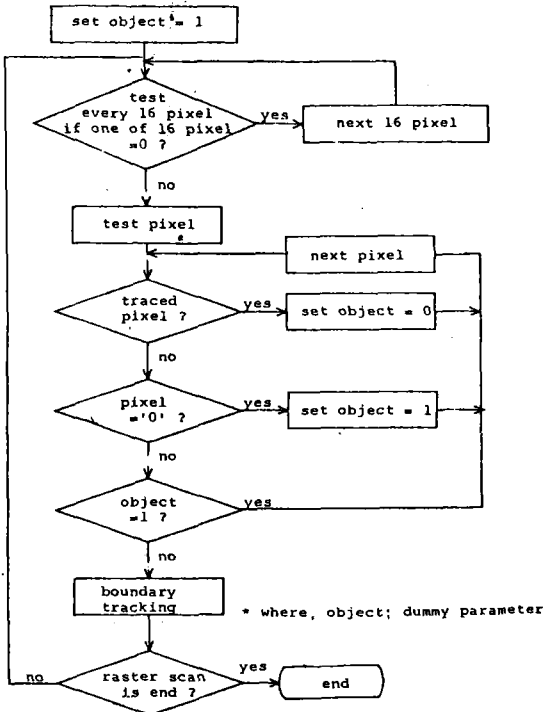


Fig. 4.1

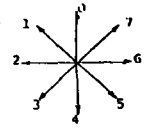
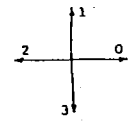


Fig. 4.2

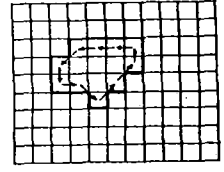
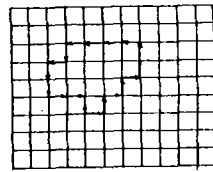


Fig. 4.3

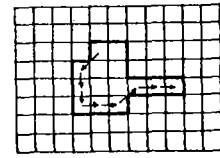
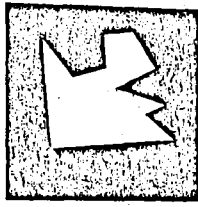
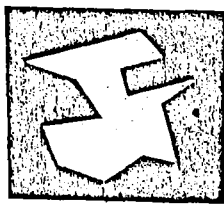
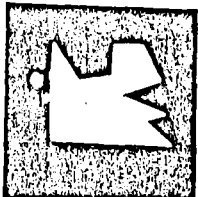
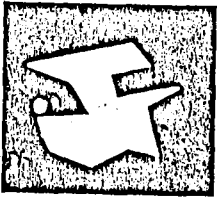


Fig. 4.4



a) real image



b) digitized image



c) after boundary tracking

Fig. 6.4