

한 이미지 평면에서 다물체 위치의 실시간 화상처리 알고리즘 개발

Development of Real-Time Image Processing Algorithm on the Positions of Multi-Object in an Image Plane

장완식*[†], 김정석*, 이성민**
 W. S. Jang*[†], K. S. Kim* and S. M. Lee**

초 록 본 연구는 속도 향상을 고려한 실시간 다물체 화상처리 알고리즘을 개발하고자 한다. 최근 들어 비전시스템의 사용은 검사 및 로봇 위치 제어 등에서 급속히 증가하고 있다. 이러한 비전시스템을 적용하기 위해서는 3차원 공간상 물체의 좌표를 CCD카메라에 의해서 얻어진 이미지 정보로 변환하는 것이 필요하다. 검사 및 로봇 위치 제어 작업들에 비전시스템을 적용하기 위해서 이미지 평면에서 물체의 중심 위치를 알아야 한다. 특히, 그것의 물체 형상을 표시하기 위하여 여러 개 큐들을 사용하는 강체의 경우에는 여러 개 큐들의 각각 위치 값들이 동시에 하나의 이미지 평면에서 결정되어 져야 한다. 이러한 문제를 해결하기 위하여 여러 개 큐(다물체)에 대한 화상처리 알고리즘 개발 과정을 본 논문에서 제시하고, 개발된 알고리즘의 타당성을 제시하였다.

주요용어 : 다물체, 여러개 큐(다물체), 화상처리, 이미지 평면, 검사, 픽셀

Abstract This study is concentrated on the development of high speed multi-object image processing algorithm in real time. Recently, the use of vision system is rapidly increasing in inspection and robot's position control. To apply the vision system, it is necessary to transform the physical coordinate of object into the image information acquired by CCD camera. Thus, to use the application of the vision system to the inspection and robot's position control in real time, we have to know the position of object in the image plane. Particularly, in case of rigid body using multi-cue to identify its shape, the each position of multi-cue must be calculated in an image plane at the same time. To solve these problems, the image processing algorithm on the position of multi-cue is developed.

Keywords: multi-object, multi-cue, image process, image plane, inspection

1. 서 론

공장 자동화에 이용되는 로봇 센서의 비접촉센서인 CCD 카메라를 이용한 비전 시스템은 처리할 데이터의 양이 방대하며 제어기법의 개발이 복잡하고 어려워 지금껏 많은 관심을 끌지 못하고 단지 제품의 검사, 주변의

감시등의 단순한 작업에만 사용되었다. 최근 3~4년 사이에 비전 시스템은 제어하고자 하는 대상을 중심으로 주변 작업환경을 통합적으로 인지하여 처리함과 동시에, 주변 작업환경의 변화에 따른 상황을 예측하여 처리할 수 있는 장점 때문에 여러 분야에서 많은 연구가 이루어지고 있다[1-4]. 비전 시스템의 개발 과정을 살펴보면 1960

년대 후반에 Stanford 대학의 McCarthy에 의해 TV 카메라, 로봇 그리고 컴퓨터로 이루어진 비전 시스템이 처음으로 개발되었다. 1973년에는 물 펌프 조립작업에 실질적으로 비전 시스템이 적용되었으며, 1980년대 후반부터는 화상처리 등의 센싱 기술을 로봇에 적용시켜 사람을 대신하여 로봇을 산업현장에 배치하고자 하는 노력들이 시도되기 시작했다[5-9].

그러나 비전 센서를 이용한 로봇 비전 시스템을 실제 산업현장에 적용하기까지는 해결해야 할 문제점은, 많은 데이터를 저장하기 위한 메모리의 확보와 메모리 손실에 따른 처리속도의 저하이다.

이러한 문제를 해결하기 위해 최근 들어 많은 연구가 이루어져 상당한 개선을 이루었다. H/W적인 부분에서 1970년대 후반부터 이루어진 LSI(Large Scale Integrate)로 대표되는 대규모 기억소자의 꾸준한 개발로 인해 메모리 문제를 해결하였다. 아울러 마이크로 프로세서의 발달에 의하여 많은 양의 데이터에 의한 처리 속도의 저하와 복잡한 알고리즘에 의한 처리 속도의 저하에 대한 문제가 상당히 해결되었다.

Bowerman은 한 이미지 평면상에 존재하는 한 개의 물체가 여러 개의 픽셀에 걸쳐 존재하므로 물체 중심 위치 결정을 위하여 단면 1차 모멘트 방법을 이용한 단일 물체 화상처리 기법을 개발하여 비전 시스템을 로봇 위치 제어 등에 활용할 수 있는 가능성을 제시하였다. 또한, 이치화 기법 또는 큐(cue)를 사용하여 데이터의 양을 최소로 하며 작업의 조건과 특성을 명확히 함으로서, 비전 시스템을 특정 작업에만 적합하도록 적용하고 획득되어지는 데이터의 양을 최소로 줄이는 등의 물리적인 작업 환경을 통해 데이터의 처리소요시간을 줄이는 방법들이 연구되었다[1,7,10]. 특히 Skaar 등[7,10]은 여러 개 큐(혹은 LED)를 사용할 때 한 이미지 평면에 주어진 여러 큐에 대해서 동시에 화상처리를 실시하지 않고 각 큐에 대해 on-off 스위치를 작동하여 한 개씩 화상처리를 실시하거나, 혹은 한 이미지 평면상에 존재하는 여러 큐를 소프트웨어 방법으로 1개씩 존재하게 하여 Bowerman[11] 방법을 이용하여 각 큐에 대해 화상처리를 실시하였다. 이로 인해 한 이미지 평면상에 존재하는 모든 큐를 화상처리 하는데 많은 시간이 소요되어 실시간 응용을 어렵게 하는 단점을 갖고 있다.

지금까지 대부분 화상처리 알고리즘은 작업의 대상인 물체의 형상을 전체적으로 인식하여 처리함으로써 데이터의 양이 방대하고 데이터를 처리하는 알고리즘이 매우

복잡하였다. 그리고, 로봇에 이용되는 3차원 물체의 형상을 표현하기 위해서는 최소 3개 이상의 큐가 필요하므로, 한 이미지 평면에서 동시에 다수의 큐(다물체)에 대한 화상처리 방법이 필요하다. 이와 같은 문제점을 해결하기 위해 본 연구는 방대한 데이터 양의 감소를 위해 큐를 사용하였고, 실시간 로봇 위치 제어를 위해 한 이미지 평면상에 존재하는 여러 개 큐를 동시에 처리 할 수 있는 다물체 화상처리 기법을 개발하여 화상처리 속도를 향상시켰다. 마지막으로 실험을 통하여 개발된 기법의 타당성을 제시하였다.

2. 다물체 화상처리 기법개발

CCD 카메라의 이미지 평면상의 데이터로부터 사용자의 목적에 유용한 값을 찾아내는 작업을 화상처리라 하며, 비전 시스템의 적용에 있어 가장 기본이 되는 작업중의 하나이다.

여기서, 하나의 이미지 평면에서, 다물체 위치의 인식을 위해 Fig. 1과 같은 다물체 화상처리 기법을 개발하였다.

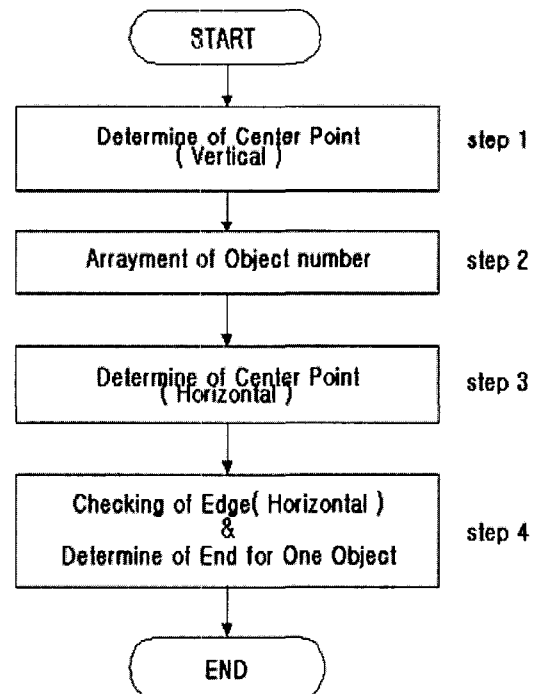


Fig. 1 Block diagram of image processing for multi-object

다음 각 절은 Fig. 1 블록선도의 4단계에 대한 각각의 알고리즘을 설명한다.

2.1. step 1 : 수직방향의 중심 계산

각 물체 수직 방향의 중심을 계산하는 과정이다. 여기서 가장 중요한 문제는 물체들이 Fig. 2과 같이 겹쳐서 존재하는 경우이다. 이 경우 Object 1과 Object 2의 각각에 대해 수직방향의 중심(vertical center)값을 처리해야 한다.

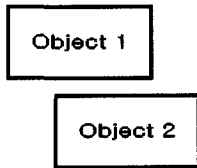


Fig. 2 Positions of objects in image plane

위 문제의 해결을 위해 이미지 평면이 한 행 단위로 스캔된다는 조건을 두고 Fig. 3과 같은 방법을 개발하였으며 각 블록에 대한 설명은 다음과 같다.

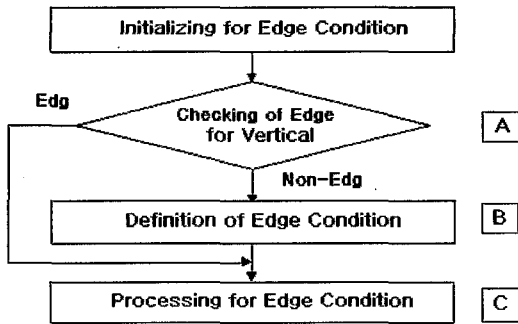


Fig. 3 Block diagram for step 1

- [A] - Graylevel을 이용하여 물체의 경계를 검사하기 위해 $|pixel(n) - pixel(n-1)| > a$ 을 만족하면 물체의 경계로 인정
- [B] - 물체의 내부와 외부를 판별하기 위하여 변수 nn을 사용하여 경계번호 부여하고 물체의 시작과 끝을 새로운 변수를 사용하여 경계값 (변수명: compare) 에 저장
- [C] - convict라는 변수를 두어 물체의 끝을 인식한다. 이에 대한 자세한 내용은 Fig. 4 에서 보여주며 각 블록에 대해 설명하면 다음과 같다.

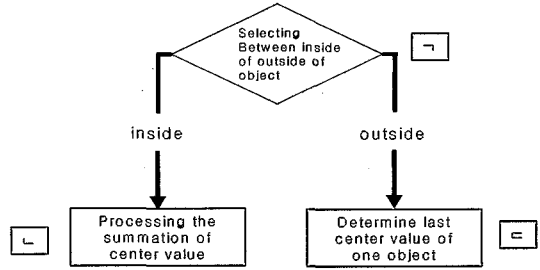


Fig. 4 Block diagram for C in Fig. 3

여기서,

- 블록 [L]은 물체의 내부와 외부로 검사 (nn이 홀수이면 물체의 내부, nn이 짝수이면 물체의 외부),
- 블록 [E]은 내부인 경우 누적 알고리즘 적용($yv=yv+j$, $count = count+1$)을 나타내며,
- 블록 [E]은 nn이 짝수고 convict가 1로 지정되어 있을 경우에만 실행되며 수직방향의 중심점을 구하여 저장한다.

2.2. step 2 : 물체의 번호 정렬

전 단계의 데이터와 현 단계의 데이터 상호간의 관계를 통해 현 단계의 물체와 전 단계의 물체간의 종속관계를 정의할 수 있다. 물체는 일반적으로 Fig. 5 과 같은 물체 형상으로 구성되며 식(1)을 사용하여 단일 물체인지 이중의 물체인지를 구분할 수 있다.

- 형상 1인 경우;

$$compare_{2n-1} \leq compare_{2n-1}$$

$$AND \quad compare_{2n} \leq compare_{2n}$$
 - 형상 2인 경우;

$$compare_{2n-1} \geq compare_{2n-1}$$

$$AND \quad compare_{2n} \geq compare_{2n}$$
 - 형상 3인 경우;

$$compare_{2n-1} \leq compare_{2n-1}$$

$$AND \quad compare_{2n} \geq compare_{2n}$$
 - 형상 4인 경우;

$$compare_{2n-1} \geq compare_{2n-1}$$

$$AND \quad compare_{2n} \leq compare_{2n}$$
- (1)

여기서,

- $compare_{2n-1}$: 전 단계 물체가 시작하는 값.
- $compare_{2n}$: 전 단계 물체가 끝나는 값.

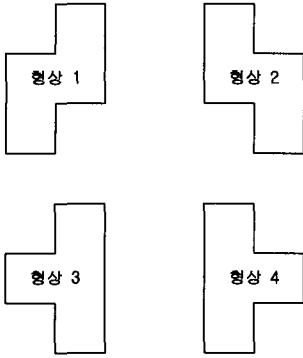


Fig. 5 Example for feature of objects

식(1)을 만족할 경우 현 단계 물체에게 종속관계에 있는 전 단계의 물체 번호(object number)를 부여하고, 식(1)을 만족하지 않을 경우 현 단계의 물체에게 전 단계가 가지는 (최대물체번호+1)의 값을 물체번호로 부여함으로써 물체번호를 정렬한다. 이를 구현하기 위해서 Fig. 6과 같은 방법을 개발하였

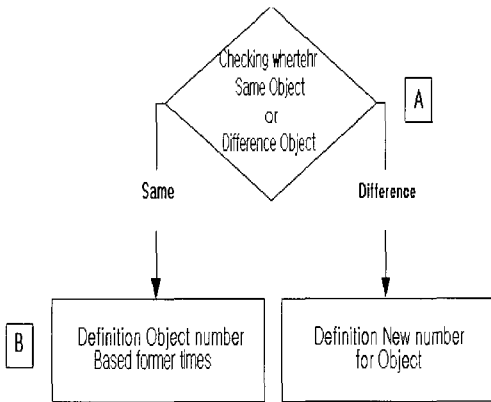


Fig. 6 Block diagram of step 2 in Fig. 1

여기서,

- Ⓐ : 식(1)의 조건을 통해 현 단계의 물체와 전 단계의 물체간의 종성의 검사
- Ⓑ : 식(1)을 만족하는 현 단계의 물체는 전 단계의 물체 번호의 부여
- Ⓒ : 식(1)을 만족하지 않을 경우 현 단계의 물체번호에 전 단계가 가지는 (최대물체번호+1)의 값 부여

2.3. step 3 : 수평방향의 중심점 계산

부여된 물체번호에 따라 수평방향의 값을 합하고 이를 수평데이터의 개수로 나누어 수평방향의 중심점을 계산한다.

2.4. step 4 : 물체의 종료검사

처리가 완료된 단일 물체를 인식하여 처리하는 과정이다. 전 단계 물체의 시작과 끝의 범위에서 $|pixel(n-1) - pixel(n)| \leq a$ 을 한번이라도 만족한다면 물체는 계속되어짐을 나타내고, 만족하는 경우가 한번도 없을 경우는 그 물체의 최종 결과를 저장하고 그 물체보다 물체번호가 나중인 물체들에게 (물체번호-1)이라는 새로운 물체번호를 부여한다.

이를 구현하기 위하여 Fig. 7과 같은 방법을 개발하였다.

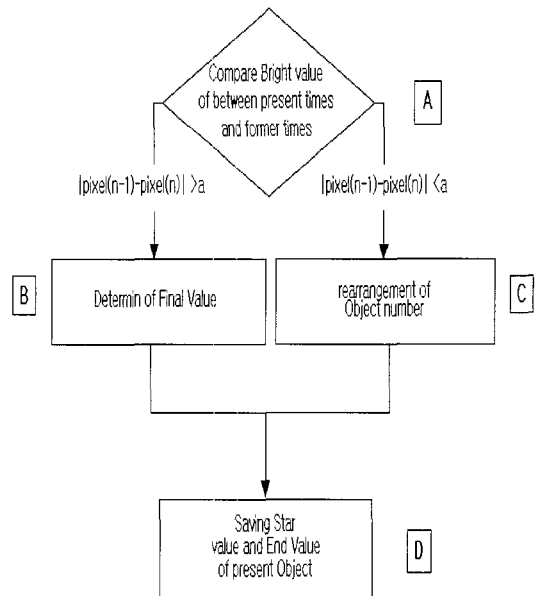


Fig. 7 Block diagram of step 4 in Fig. 1

여기서,

- Ⓐ : 전 단계 물체가 시작하는 값 ($comparel[2n-1]$)과 종료값 ($comparel[2n]$)의 범위에서 $|pixel(n-1) - pixel(n)| \leq a$ 를 이용하여 전 단계의 gray level 과 현 단계의 gray level 비교

- ⓑ : ⓐ의 조건을 한번도 만족하지 않은 경우 단일 물체는 종료된 것으로 인식하고 그 물체의 최종값 도출
- ⓒ : ⓐ의 조건을 한번이라도 만족할 경우는 물체가 연속된 것으로 인지하고 ⓑ조건을 만족한 다음의 물체로부터 (물체번호-1)을 새로운 물체번호의 적용
- ⓓ : ⓐ에서 사용하기 위한 물체의 시작값과 종료값 저장

지금까지 다물체 화상처리 기법을 개발하는 과정을 설명하였으며, 여기서 물체의 경계부분을 구별하기 위해서 사용되는 $| pixel(n-1) - pixel(n) | \leq a$ 에서 a값을 60으로 설정하였다. 그 이유는 본 연구의 실험이 실험실에서 행해지기 때문에 실험실 안의 조명이나 채광 등의 영향으로 인해 물체가 아닌 부분에도 상당한 gray level값이 갖게 되므로 이를 보완하기 위해 수차례 예비 실험을 통하여 물체 경계값을 나타내는 최적의 a값으로 60을 사용하였다.

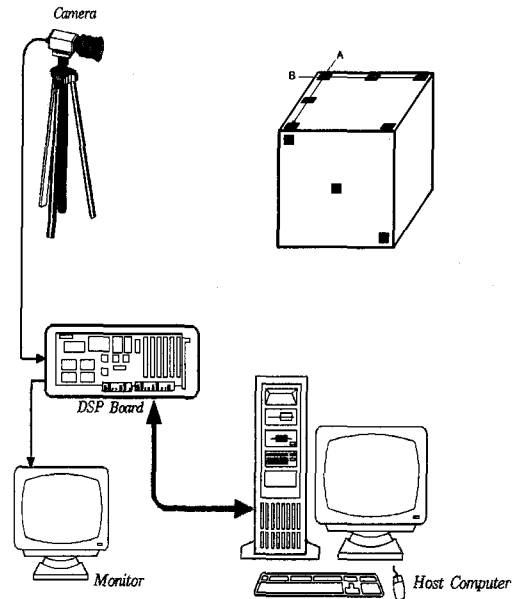


Fig. 8 Experimental set-up

3. 실험장치 및 절차

3.1. 실험장치

화상처리 실험은 단일 물체 화상처리 실험과 복합 물체 화상처리 실험으로 구성되며, 실험장치는 Fig. 8에 나타내었으며 각 구성요소의 사양은 다음과 같다.

(1) 비전 시스템

- (a) CCD 카메라(1대) :
 - 동경전자 공업주식회사 제작
 - AVENIR TV 줌 렌즈를 부착
 - 768(H)×494(V)의 유효 화소수
- (b) 화상처리 비전 보드 :
 - 삼성전자 제작(FARA MVB-03, 흑백)
 - CPU - TMS320C31(33MHz)
 - 해상도 - 640(H)×480(V)
 - 화상도 - 256계조
 - Frame - 4 Frames(2banks by 2frames)
 - LUT - 16 banks
- (c) 모니터 : 동양 정밀공업주식회사(OPC)의 12" 흑백 비디오 모니터
- (d) Host PC : 삼성 펜티엄 II PC(400MHz)

3.2. 시험모형

다중 물체 위치결정을 위한 화상처리 실험을 위해서 150×150×147(mm)크기의 Fig. 9와 같은 직육면체의 블록을 제작하고 수직, 수평, 복합적인 각각의 경우에 대한 실험을 위해 큐를 부착하였다. 여기서 A는 수직 경우의 실험을 위한 큐의 배열이며 B는 수평 경우의 실험을 위한 큐의 배열이다. 복합적인 경우의 실험을 위해서는 시험모형에 부착되어 있는 7개 큐를 사용한다.

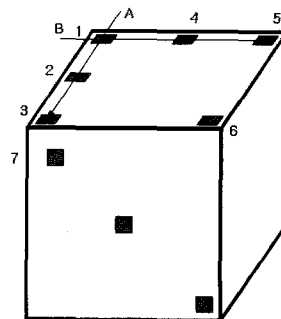


Fig. 9 Test model for experiment of image processing

3.3. 시험절차

모든 비전 시스템의 처리를 위하여 화상 처리가 기초가 되고 3차원 물체를 인식하기 위해서 한 이미지 평면의 3점 이상의 데이터처리가 필수적이다.

본 실험은 3차원 물체의 형상을 인식하기 위하여 3점 이상의 물체에 대한 화상정보를 동시에 처리할 수 있도록 개발한 다중 화상처리 기법의 타당성을 확인하는 것이다.

본 실험은 Fig. 10에서 보여주는 절차에 따라 수행된다. MVB-03는 FM에 있는 화상 데이터를 스캔하기 위한 것으로 FM_rVerLine 명령어를 통해 수직방향으로 스캔하는 방법과 FM_rHorLine 명령어를 통해 수평방향으로 스캔하는 방법을 제공한다. 본 실험에서는 메모리를 적게 차지하고 처리속도가 좀더 빠른 FM_rVerLine 명령어를 사용하여 프로그램 하였다. 따라서 먼저 수직방향 데이터가 올바르게 처리되어야 수평방향 데이터의 올바른 처리를 기대할 수 있고, 수직·수평 방향 데이터가 올바르게 처리되어야 복합적인 경우에 대해서도 올바른 데이터 처리를 기대할 수 있다.

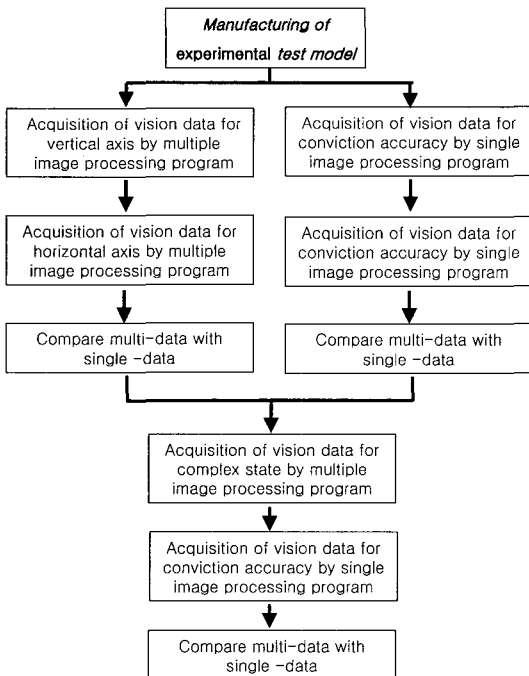


Fig. 10 Experiment procedure for multiple image processing

Fig. 9의 시험모형에서 수직방향을 위한 큐 배열 A를 부착하여 복합 물체를 위한 비전 데이터를 획득한 후, 각각 큐의 위치를 비교하기 위하여 각각 큐의 비전 데이터를 획득한다. 수평방향을 위한 큐 배열 B를 부착하고 수직방향에서 사용한 방법을 사용하여 각각의 비전 데이터를 획득한다. 그리고 복합적인 경우에도 위의 방법을 그대로 이용한다.

4. 실험결과

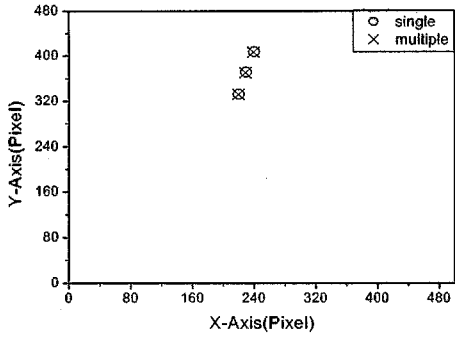
2점 이상의 물체에 대해 화상을 동시에 처리하기 위한 기본실험으로 매우 단순한 수직인 경우(시험모형의 A선)와 수평인 경우(시험모형의 B선)를 실험 모형에 설정하였으 며, 각각 3큐를 부착하여 화상처리를 실시하였다. Table 1은 본 연구에 개발된 다물체 화상처리 알고리즘의 결과와 단일물체 화상처리 방법[11]에 의해서 얻어진 결과를 나타내준다.

Fig. 11 과 Fig. 12 는 Table 1의 결과를 그래픽 적으로 보여주며, 각 Fig의 (a)는 수직 및 수평인 경우에 대해 본 연구결과와 단일 물체 화상처리 방법에 의한 결과를 비교한 것을 보여주며, (b)는 두 방법에 의해 얻어진 결과에 대한 오차를 나타낸다. 각 Fig의 (b)에서 보여준 수직과 수평인 경우, 3개의 큐에 대한 화상처리 오차는 단일물체 화상처리 기법과 비교할 때 ±1.5 픽셀 이내로 매우 만족할 만한 값을 얻었다. 또한, 각 경우에 단일물체 화상처리 기법을 사용 시 화상처리 시간은 각 큐에 대해 0.8 s가 소요되어 3개 큐를 처리시 약 2.4 s가 소요되었다. 반면에 본 연구에서 제시된 기법은 3개 큐를 동시에 처리하는데 약 1.2 s가 소요되어 약 2배 정도의 속도 향상을 보였다.

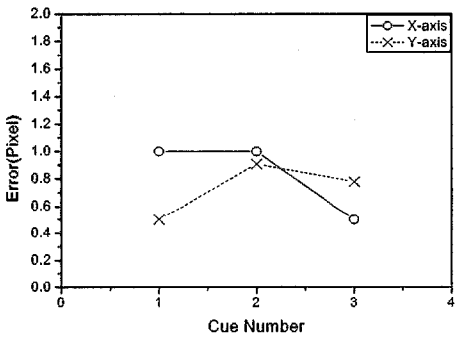
Table 1 Image processing data in vertical & horizontal direction

(unit : pixel)

		Ver(A-line)		Hor(B-line)			
	Cue No.	X-Axis	Y-Axis	Cue No.	X-Axis	Y-Axis	
1	Single	240.00	407.31	1	Single	240.00	407.31
	Multiple	239.00	406.81		Multiple	239.00	406.82
2	Single	229.50	371.43	2	Single	338.50	408.86
	Multiple	228.50	370.52		Multiple	337.00	408.47
3	Single	220.00	333.22	3	Single	439.19	409.84
	Multiple	219.50	332.44		Multiple	439.00	409.45



(a)

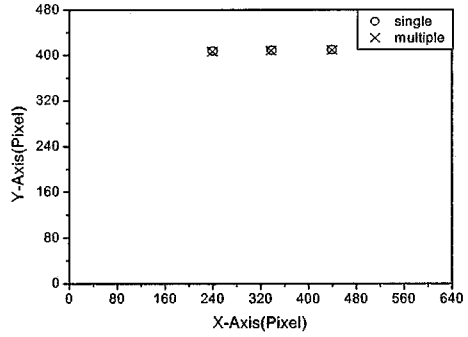


(b)

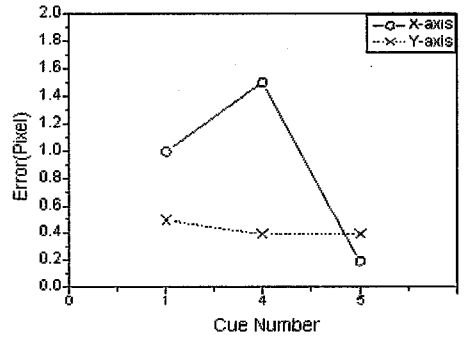
Fig. 11 Multi-cue results for processing in the vertical. (a) comparison of single and multiple algorithm, (b) Errors between single and multiple results

최종적으로 다물체의 화상처리 기법의 타당성을 증명하기 위하여 시험모형에 임의적으로 배열되어 있는 7점을 선택하여 본 연구에 개발된 다물체 화상처리 알고리즘을 적용하였다. Table 2는 본 연구 결과와 단일 물체 화상처리 방법[11]을 통해 얻어진 결과를 비교하여 나타낸 것이다.

Fig. 13 은 Table 2의 결과를 그래픽으로 보여주며, (a)는 본 연구 결과와 단일 물체 화상처리 기법의 결과를 7개 큐에 대해서 비교하여 보여주며, (b)는 두 방법의 결과에 대한 오차를 보여준다. (b)에서 보여준 것 같이 7개 큐를 사용했을 때 앞서 보여 준 3개 큐를 사용한 결과와 다르게 5~7번 큐에서는 최대 ± 7 픽셀의 오차가 발생하였으며, 이 오차는 카메라 이미지 평면의 원점을 기준으로 큐의 중심 위치를 계산하면 $\pm 1.4\%$ 가 된다. 이러한 오차가 발생한 이유는 한 이미지 평면상에 큐가 여러 개



(a)



(b)

Fig. 12 Multi-cue result for image processing in the horizontal. (a) comparison of single and multiple algorithm, (b) Errors between single and multiple results

존재했을 때, 큐 상호간에 영향을 주는 gray level이 물체를 인식하는 경계값에 영향을 끼친 것으로 생각된다. 이것은 여러 개 큐에 대해 본 연구의 예비 실험 결과 5개 큐까지는 3개 큐와 마찬가지로 최대 ± 2 픽셀이내의 오차를 갖고 있으나 6개 이상의 큐에서는 ± 7 픽셀의 오차를 갖고 있음을 확인하였다. 또한, 7개 큐에 대해 단일물체 화상처리 기법을 사용 시 화상처리 시간은 각 큐에 대해 0.8 s가 소요되어 7개 큐를 처리시 약 5.6 s가 소요되었다. 반면에 본 연구에서 제시된 기법은 7개 큐를 동시에 처리하는데 약 1.8 s가 소요되어 약 3배 이상의 속도 향상을 보였다.

이와 같이 본 연구에서 개발된 다물체 화상처리 기법에 의하여 3개 큐의 처리 소요시간을 단일 물체 화상기법에 의한 처리 소요시간과 비교할 때 2배 정도의 속도가 향상 되었고, 5개 이하의 큐에 대하여 두가지 방법에

의하여 얻은 물체 위치값은 거의 같았다. 따라서 본 연구에서 개발된 다물체 화상처리방법을 로봇 위치 제어 및 여러 응용 분야에 적용이 가능하다고 생각된다.

Table 2 Vision data of multi-cue

(Unit : Pixel)

Cue No.	X-Axis	Y-Axis	Cue No.	X-Axis	Y-Axis		
1	Single	240.00	407.31	5	Single	439.20	409.84
	Multiple	240.00	408.07		Multiple	438.00	404.51
2	Single	229.50	371.43	6	Single	430.00	332.06
	Multiple	229.00	371.49		Multiple	436.50	332.72
3	Single	220.00	333.22	7	Single	250.50	248.85
	Multiple	220.00	333.38		Multiple	250.50	241.71
4	Single	338.50	408.86				
	Multiple	338.00	409.26				

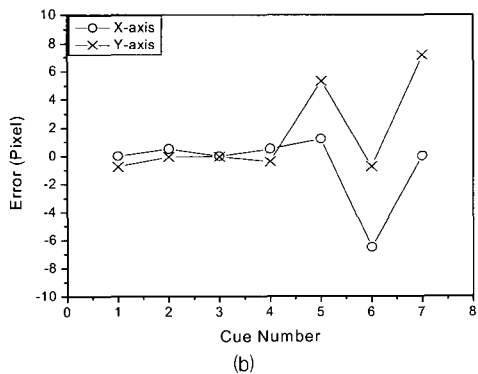
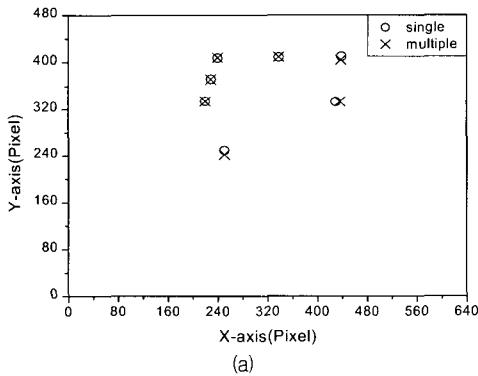


Fig. 13 Results of image processing of multi-cue. (a) comparison of single and multiple algorithm, (b) Errors between single and multiple results

5. 결 론

본 연구에서 다물체 화상 처리 기법을 개발하여 단일 물체 화상처리기법에 의한 결과와 비교분석하여 다음과 같은 결론을 얻었다.

- (1) 물체의 화상정보를 처리할 경우 물체의 경계를 추출하여 물체의 끝을 인지하고 화상처리 보드의 동작을 억제함으로써 화상처리 속도의 향상을 기하였다.
- (2) 다물체의 화상 정보를 처리하는 화상 처리 기법을 개발하였으며 다물체에 대한 화상 처리 기법의 결과 값과 단일 물체 화상처리 기법을 통하여 각각의 물체에 대해 얻은 결과 값을 비교하였을 때 거의 일치하였다.
- (3) 다물체 화상처리 기법의 처리 속도는 기존의 단일 물체 화상처리 기법과 비교하면, 3개의 큐에 대해서 2배가 향상되었고, 큐의 수가 많으면 많을수록 2배 이상 향상시킬 수 있다.
- (4) 향후 본 연구결과에 고 성능 비전 보드 및 컴퓨터를 이용하면 더 좋은 성능 향상이 기대되어, 물체 검사 및 로봇 위치제어 등에 효과적으로 이용될 수 있을 것으로 생각된다.

후 기

본 연구는 한국과학재단 목적 기초 연구(R05-2000-000-00320-0) 지원으로 수행되었음.

참고문헌

- [1] K. P. H. Berthold, Robot vision, Cambridge Massachusetts: The MIT Press, pp. 46-48, (1986)
- [2] C. H. Hong, S. M. Chun, J. S. Lee and K. S. Hong, "A Vision-Guided Object Tracking and Prediction Algorithm for Soccer Robots," Proceedings of the 1997 IEEE International Conference on Robotics and Automation: April, pp. 346-351, (1997)
- [3] W. Li and L. Grier, "An Application of On-line Calibration of Machine Vision for the Object Position Determination," The Fourth International Conference

- on Control, Automation, Robotics and Vision (ICARCV'96): 3-6 December, pp. 105-109, (1996)
- [4] S. B. Skaar, H. W. Brockman and R. Hanson, "Camera-Space Manipulation," *The International Journal of Robotics Research*, Vol. 6, No. 4, pp. 20-32, (1987)
- [5] W. Z. Chen, U. A. Korde and S. B. Skaar, "Position Control Experiments Using Vision," *Int. J. Robotics Research*, Vol. 13, No. 3, pp. 199-208, (1994)
- [6] P. I. Corke, "Video-Rate Robot Visual Servoing", In K. Hashimoto Editor, *Visual Servoing*, pp. 257-283, (1993)
- [7] J. G. Emilio and S. B. Skaar, "Efficient Camera-Space Manipulation using Moments," *IEEE International Conference on Robotics and Automation*, April, pp. 3407-3412, (1996)
- [8] E. W. Lee, C. S. Arthur and P. N. Charles, "Dynamic Sensor-Based Control of Robots with Visual Feedback," *IEEE Journal of Robotics and Automation*, Vol. 3, No. 5, October, pp. 404-417, (1987)
- [9] Peter A. S., "Control of eye and arm movements using active, attentional vision," *Applications of AI, machine vision and robotics*: pp. 1471-1491, (1993)
- [10] J. G. Emilio, S. B. Skaar and J. S. Micheal, "Efficient Camera-Space Target Disposition in a Matrix of Moments Structure Using Camera-Space Manipulation," *The International Journal of Robotics Research*, Vol. 18, No. 8, August, pp. 809-818, (1999)
- [11] E. R. Bowerman, "Video Target Locator," *United States Patent 4,044,377*, August 23, (1977)