

스테레오 비전을 이용한 Bin Picking Method

° 주기세, 한민홍
고려대학교 산업공학과

Bin Picking Method using Stereo Vision

° Kisee Joo, Min-Hong Han
Department of Industry, Korea University

ABSTRACT

This paper presents a Bin-Picking method in which robot recognizes the positions and orientations of jumbled objects placed in a bin, then picks up distinctive objects from the top of the jumble. The jumbled objects are recognized comparing the characteristics extracted from stereo images with those in the CAD data.

The 3-D information is obtained using the bipartite-matching method which compares image of one camera with the image of the other camera. Then the robot picks up the object which will cause the least amount of disturbance to the jumble, and places it at a predetermined place.

This paper contributes to the basic study of Bin-Picking, and can be used in an automatic assembly system without using part sorting or orienting devices.

위와 같은 문제점의 해결책으로서 Bin-Picking이 제시 되었는데, Bin-Picking이란 물체가 무작위로 쌓여 있더라도 로봇이 제일 위에 있는 물체부터 하나씩 집어서 원하는 곳에 위치시키는 것을 의미한다. 무작위로 쌓여 있는 물체들을 인식하기 위해서는 image processing, image segmenting, image modeling이 필요하게 되는데 불분명한 대조, 왜곡현상, 노이즈등 때문에 정확한 정보를 얻기가 어렵기 때문에 CAD data, max curvature, texture 등 특정점 및 물체형상 비교를 이용해서, 정확한 정보를 얻기 위한 노력들이 계속되어져왔다.

이전의 경우는 대부분이 단순한 경우인 conveyor belt 상에서나 작업장에서, 서로 분리된 물체인식이 대부분을 이루고 있다. 물체가 서로 겹치지 않고 분리된 상태에서 의 물체인식에 대한 연구는 Tsui et al(1973), Mckee and Aggarwal(1977)에 의해서 연구되었으며, Baird(1977)는 two light planes의 사용에 의해서 silhouettes를 추출하는 "Consight" vision system를 개발했다. 또한 Rosen et al (1974)는 conveyor belt상에 있는 서로 분리된 parts들을 들어 올리는 system을 개발했다.

그러나 본 논문에서는 대칭이면서, 특정점을 찾기 힘든 물체가 쌓여 있는 경우에, CAD정보, concave point, orientation, distance정보를 이용하여 image를 segmenting 시키고, weight 계산후 optimality test를 이용해서 image modeling을 시키고 난후, bipartite matching에 의한 camera-1 및 camera-2 의 corresponding object 및 vertex를 식별한 후 3D정보를 구하고, 물체가 서로 교차하는 경우, 물체의 교차점을 구한후 교차점에서의 3D정보를 구함으로써, 제일 위에 있는 물체부터 순서적으로 robot가 작업을 하게된다.

1. 서론

부품들로 쌓여져 있는 bin이나 container는 잘 정리된 경우, 부분적으로 정리된 경우, 무작위로 쌓여져 있는 경우 등으로 분류할 수 있는데, 지금까지 대부분의 공장에서는 로봇이 부품을 쉽게 잡을 수 있도록, 일정한 방향 및 간격으로 잘 정리된 상태의 부품이 로봇에 제공되어져 왔다.

그러나 부품들을 정리하기 위해서는 작업자의 수작업이 필요하고, 이와같은 수작업은 대부분이 열악한 작업 환경 및 위험성 때문에 산업재해가 빈발하는 작업중의 하나였다. 또한 로봇이 물체를 쉽게 잡을 수 있도록 물체의 방향을 자동적으로 제공하기 위해서 feeding device, orienting device 등이 부품의 외형에 적합하게 제작되어 현장에서 사용되고 있으나, 부품 외형의 손상, 적용의 한계성 등 적용하는데 많은 제약이 따른다.

2. 좌표계 시스템

2.1 Stereo Camera Modeling

Stereo vision이란 두 점 또는 그이상의 다른점들로

부터 3차원 정보를 얻는 것을 의미한다. 인간의 두 눈과 유사하게 두 camera를 나란히 위치시키는 시스템을 lateral model이라고, 또 인간이 보이지 않는 물체를 보기위해서 고개를 돌거나 몸을 움직이는 것과 같은 형태를 본 떠 만든 것을 axial motion model이라한다.

2.2 좌표계 시스템

로보트가 정해진 위치 및 방향에 있는 물체를 들어 올리기 위해서는 그림 1과 같은 카메라 좌표계, 화면 좌표계, 월드 좌표계, 로보트 좌표계등 4개의 좌표계 시스템을 정의하여 사용한다.

첫번째 좌표계인 카메라 좌표(x_c, y_c, z_c)는 카메라의 시축을 z_c 축으로 하고 x_c 축과 y_c 축은 z_c 축에 따라 왼손법칙에 의해서 결정된다. 두번째 좌표계인 화면 좌표계(x_w, y_w)는 카메라로 잡은 실제계상의 영상을 비전시스템으로 처리하기 위하여 설정되는 좌표계이며, 화면상의 중앙점에 해당하는 image center를 원점으로 하고, 화면에 수평하게 x_w 축을 잡고, x_w 축에 수직인 축을 y_w 축으로 정의한다.

세번째 좌표계인 월드좌표(x_w, y_w, z_w)는 평면과 카메라 시축이 가상평면에 사영된 선을 y_w 축으로 하고, y_w 축에 수직인 축을 x_w 축으로 하고, x_w, y_w 평면에 수직인 축을 z_w 축으로 한다. 네번째 좌표계인 로보트 좌표(x_r, y_r, z_r)는 x_r, y_r, z_r 축은 x_w, y_w, z_w 축과 동일하게 정의하고 단지 원점을 월드좌표계의 원점으로부터 Δx 만큼 평행이동 시킨 점을 원점으로 정의한다.

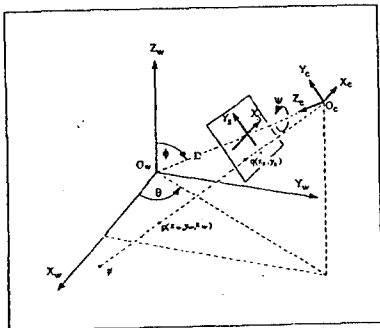


그림 1 좌표계 시스템

2.3 각 좌표계에서의 좌표값 계산

물체의 위치 및 방향을 계산하기 위해서는 좌표계 사이의 관계를 이해할 필요성이 있다. 즉 화면좌표로부터

카메라좌표를 구하고, 카메라 좌표로부터 월드좌표를 구하는 과정을 나타내면 아래 두 식 (2-3-1), (2-3-2)식으로 표현 가능하다.

$$\begin{aligned} x_c &= x_w * z_c / d \\ y_c &= y_w * z_c / d \end{aligned} \quad (2-3-1)$$

여기서 x_w, y_w, z_w 는 카메라 좌표, x_c, y_c 는 화면좌표 d 는 focal length

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \begin{pmatrix} x_w & y_w & z_w & 1 \end{pmatrix} \begin{pmatrix} -\sin\theta & -\cos\theta\cos\phi & -\cos\theta\sin\phi & 0 \\ \cos\theta & -\sin\theta\cos\phi & -\sin\theta\sin\phi & 0 \\ 0 & \sin\phi & -\cos\phi & 0 \\ 0 & 0 & d & 1 \end{pmatrix} \quad (2-3-2)$$

여기서 x_w, y_w, z_w 는 월드좌표 x_c, y_c, z_c 는 카메라좌표 d 는 focal length, θ 는 pan각, ϕ 는 tilt각

화면좌표상의 한점(x_w, y_w)의 역투영변환한 점과 월드 좌표계로 설정한 평면이 만나는 점을($x_w, y_w, 0$)이라 할 때 점P의 y_w 좌표 C와, 월드좌표의 원점 O_w , 카메라 좌표계의 원점 O_c , 3점이 이루는 삼각형은 그림 2로 나타내어진다.

$$z_c : D = (\ell_1 + \ell_2) : \ell_1$$

$$z_c = (\ell_1 + \ell_2) * D / \ell_1$$

여기서 ℓ_1, ℓ_2 는 아래와 같이 계산된다.

$$a = \tan^{-1}(y_w / D) \quad \ell_1 = D / \cos a$$

$$\ell_1 + \ell_2 = h / \cos(\theta + \phi)$$

위에서 계산된 ℓ_1, ℓ_2, a 를 이용하여 z_c 는 다음과 같이 결정된다.

$$z_c = \cos(a) * h / \cos(\theta + \phi)$$

일단 z_c 가 결정되면 focal length d 와 화면 좌표계의 좌표(x_w, y_w)와의 비례식을 이용해서 x_c, y_c 는 식 (2-3-1)에 의해서 결정된다.

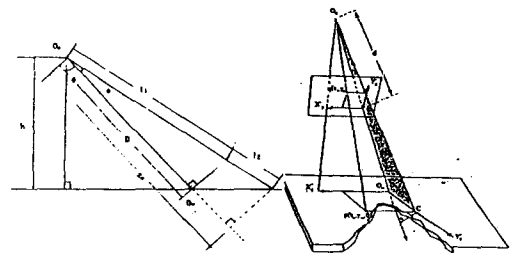


그림 2 화면좌표와 월드좌표의 관계

3. Image Modeling

본 연구에서는 연구 대상으로 분필을 사용함으로써 좌우가 서로 대칭이고, 특징점 추출이 힘들기 때문에, 정확한 물체 인식 및 분리를 위하여 CAD 정보를 이용해서 대응점을 결정짓고, 물체가 겹쳐진 부분에서의 교점은 오목점(concave point)이라는 성질, 겹쳐진 점과 이웃하는 점과의 slope, 연속하는 두 점의 거리정보를 이용해서, 물체의 특징점을 grouping하고 이것을 기준으로 CAD data의 확장, 이동, 회전에 의한 대응꼭지점들을 찾으므로써 물체들을 서로 분리 하게된다. 또한 물체가 대칭형이기 때문에 양쪽에서, 특징점을 잡는 경우와, 노이즈 및 사영에 의한 외곽현상 때문에 다른 꼭지점과 대응되는 경우에 물체의 수를 실제 갯수 보다 많게 예측한다. 따라서 정확한 image modeling을 위해서 weight 계산후, optimality test를 한다.

3.1 볼록점(convex point) 및 오목점(concave point)의 계산

물체가 볼록형(convex)이면 물체내의 모든 점은 볼록점(convex point)이고, 만약 볼록형 물체가 서로 겹치면 겹치는 부분의 교점(vertex)은 오목점(concave point)인 성질을 가지고 있다. 세개의꼭지점 $p_{i-1}(x_{i-1}, y_{i-1})$, $p_i(x_i, y_i)$, $p_{i+1}(x_{i+1}, y_{i+1})$ 이 있을 때 오목점과 볼록점은 아래식에서 구할 수 있다.

$$D = \begin{vmatrix} x_{i-1} & y_{i-1} & 1 \\ x_i & y_i & 1 \\ x_{i+1} & y_{i+1} & 1 \end{vmatrix} \quad (2-3-3)$$

$$= (x_i - x_{i-1})(y_i - y_{i-1}) - (x_{i+1} - x_{i-1})(y_i - y_{i-1})$$

여기서 만약 $D > 0$ 이면 볼록점
 $D = 0$ 이면 같은 직선 상 또는 평행
 $D < 0$ 이면 오목점

3.2 특징점(Characteristic point)추출

물체의 특징점을 찾기 위하여 연속하는 두 점이 볼록점(convex point)이고 연속점 사이의 거리가 임계값(threshold)이내이면 연속하는 두 점을 grouping시키고, CAD data를 확장 및 축소시키기 위한 방향을 결정짓기 위하여 이웃하는 오목점과의 기울기를 구한다. 위에서의 임계값은 CAD data의 위치에 따른 단축의 길이로부터 구한 값이다. 위에서의 거리, 방향, 임계값을 수식으로 표현하면 아래와 같다.

$$\text{distance} = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}$$

여기서 i 는 볼록점, $i-1$ 는 볼록점

$$\text{slope} = \tan^{-1}\left(\frac{x_i - x_j}{y_i - y_j}\right)$$

여기서 i 는 볼록점, j 는 오목점

$$\text{Threshold} = \sqrt{(x'_i - x'_{i-1})^2 + (y'_i - y'_{i-1})^2}$$

여기서 $i, i-1$ 은 CAD data상에서의 단축의 이웃점이다

3.3 CAD data를 이용한 Pattern Matching

물체가 쌓여 있기 때문에 tilt각, pan각, 높이에 따라서 물체 크기의 왜곡현상이 발생한다. 따라서 월드좌표에서의 CAD data를 식(2-3-1), (2-3-2)의 역변환을 이용해서, 화면좌표로 변환 시킨 후 grouping점으로 이동 및 회전 시키고 나서, 확대 및 축소 시키면서, CAD data의 각 꼭지점의 화면좌표와, 실제 쌓여있는 물체의 화면좌표의 꼭지점과의 거리를 계산한 후, 거리가 임계값 이내이면 대응점으로 분류하고, 일정한 길이만큼 확대 및 축소후에도 대응점을 찾지 못하면, 초기의 CAD data값을 grouping점으로 이동 및 회전 시킨 직후의 값과, 일정한 길이만큼 확대 및 축소시킨 후 그값의 중간값을 취한다.

정확한 물체분리를 위하여, 각 물체 사이의 weight를 계산한 다음, weight가 0.25, 0.5, 0.75인 물체에 대하여 optimality test를 한다.

이때 한물체와 모든 다른 사이의 weight가 0인 경우는, 한쪽 방향에서만 matching된 것이고, 다른 물체와 완전히 구별되기 때문에 optimality test에서 제외 시킨다. 또한, weight가 1인 경우는 양방향에서 완전히 matching된 것이기 때문에 optimality test에서 제외 시킨다.

3.3.1 Weight 할당 및 optimality test

한물체의 꼭지점과, 다른 물체의 꼭지점과의 matching 수에 따라서, $w_x = 0$, $w_y = 0.25$, $w_z = 0.5$, $w_w = 0.75$, $w_\theta = 1$ 로 weight을 할당한다.

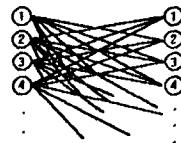


그림 3 optimality test

위에서 weight 계산후 weight가 0.25, 0.5, 0.75인 물체들의 수가 2M이라면 실제 물체의 수는 M이 된다. 따라서 weight가 0.25, 0.5, 0.75인 물체에 대한 optimality test를 하게 되는데, 변수 o_M 은 만약 물체 k가 물체 1에 matching되면 $o_M=1$, matching이 안되면 $o_M=0$ 으로 정의 된다. 변수 d_M 은 물체 k의 꼭지점들과 물체 1의 꼭지점들간의 거리의 합으로써 정의 된다.

그림으로 나타내면 그림 3과 같고 mathematical

modeling은 아래와 같이 표현될수 있다.

$$\text{Minimize } \sum_{k=1}^M \sum_{l=1}^{2M} d_{kl} * o_{kl}$$

$$d_{kl} \geq 0, \sum_l o_{kl} = 1 \text{ for } k = 1, \dots, M,$$

$$l=1, \dots, 2M, o_{kl} = 0 \text{ or } 1$$

$$\text{여기서 } d_{kl} = \sum_{i=1}^{objno} \sum_{j=1}^{objno} \sqrt{(x_M - x_U)^2 + (y_M - y_U)^2}$$

$$\text{Minimize } \sum_{k=1}^{objno} \sum_{l=1}^{objno} d_{kl} * o_{kl}$$

$$\sum_l o_{kl} = 1 \text{ for } k = 1, \dots, objtno$$

$$\sum_k o_{kl} = 1 \text{ for } l = 1, \dots, objtno$$

$$o_{kl} = 0 \text{ or } 1$$

$$\text{여기서 } d_{kl} = \sum_{i=1}^{objno} \sum_{j=1}^{objno} \sqrt{0.2 * (x_M - x_U)^2 + 0.8 * (y_M - y_U)^2}$$

3.4) Bipartite Matching에 의한 3D좌표 계산

한대의 카메라 영상으로부터 x_w, y_w 좌표는 식 (2-3-1) 및 식(2-3-2)에 의해서 계산이 가능하다. 따라서 각 물체의 높이를 알아내기 위해서는 camera-1, camera-2의 영상으로부터 분리해낸 물체중에서 어느 물체가 서로 대응하는 물체인지를 분류하고, 다음 단계로 서로 대응하는 꼭지점을 찾는 과정을 bipartite matching 방법을 사용한다.

Lateral model에서 camera-1, camera-2가 x축에 나란히 놓여있으면 camera-1, camera-2에서 계산된 x값은 서로 차이가 있지만 y값은 거의 차이가 없다.

따라서 heuristic에 의해서 y값에 의한 거리의 차에 가중치를 80%로하고 x값에 의한 거리의 차에 가중치를 20%로 한다. 그림으로 나타내면 아래와 같다.

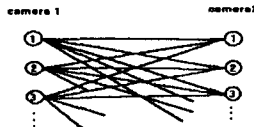


그림 4 bipartite matching

3.4.1) Bipartite Matching Method

먼저 서로 대응하는 물체를 찾은 다음에 대응하는 물체에 대한 대응점을 찾는 mathematical modeling은 아래와 같이 표현가능 하다.

1) 서로 대응하는 물체를 찾는 modeling

변수 o_{kl} 은 만약 물체 k가 물체 l에 matching되면 $o_{kl}=1$, matching이 안되면 $o_{kl}=0$ 으로 정의되며, 변수 d_{kl} 은 물체 k의 꼭지점들과 물체 l의 꼭지점들 간의 거리의 합으로써 정의 된다.

1) 서로대응하는 꼭지점을 찾는 modeling

변수 v_{ij} 는 서로 matching된 camera-1, camera-2의 물체에서 꼭지점이 서로 matching되면 $v_{ij}=1$, matching이 안되면 $v_{ij}=0$ 으로 정의하고, 변수 δ_{ij} 는 서로 matching된 camera-1에서의 물체의 꼭지점과 camera-2에서의 물체의 꼭지점 사이의 거리로서 정의 된다.

$$\text{Minimize } \sum_{i=1}^{objno} \sum_{j=1}^{objno} \delta_{ij} * v_{ij}$$

$$\sum_j v_{ij} = 1 \text{ for } i=1, 2, 3, 4$$

$$\sum_i v_{ij} = 1 \text{ for } j=1, 2, 3, 4$$

$$\delta_{ij} \geq 0, v_{ij} = 1 \text{ or } 0$$

$$\text{여기서 } \delta_{ij} = \sqrt{0.2 * (x_i - x_j)^2 + 0.8 * (y_i - y_j)^2}$$

bipartite matching에 의해서 camera-1, camera-2의 대응 물체가 분류되고, 서로간의 대응 꼭지점이 분류되면, 이미 각 물체의 x_w, y_w 좌표를 알고 있기 때문에, camera-1, camera-2의 위치와 월드좌표상의 임의의 한 점으로부터, 카메라1, 2에 대하여, 2개의 직선방정식을 구할수 있으며, 이 두 직선의 convex combination에 의해서 z_w 값을 얻을 수 있다.

4. 가장 높은 물체의 인식

만약 로봇트가 밑에 가려진 물체를 잡으려 한다면, 쌓여진 물체가 흩으려 짐으로서 물체의 위치 및 방향이 변하게 된다. 따라서, 쌓여있는 물체중에서, 제일 위에 있는 물체부터, 순서적으로 로봇트가 들어 올려야만 한다. 그러므로, 공간상에서의 기하학적 관계에 의해서, 평면과 평면 사이의 관계를 고려 해야만 한다.

각 물체를 각 물체의 장축방향을 기울기로 하고, 양 단축의 중앙을 지나는 선분으로 표시한다. 그림 5에서 점E는 camera-1의 위치를 나타내며, 선분AB는 선분PQ들과의 비교를 위한 기준 선분이며, 선분2는 공간상에서 선분AB와 서로 교차하는 경우이고, 선분1은 선분AB를 기울기 방향으로 확대 시켰을 때 서로 교차하는 경우이며, 선분3은 선분AB와 서로 교차하지 않는 경우를 나타낸다.

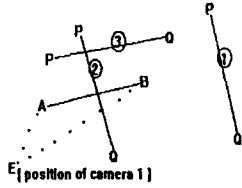


그림 5 공간상에서의 기하학적 관계

평면 EAB의 평면 방정식은 다음과 같다.

$$\begin{vmatrix} x & y & z \\ x_a & y_a & z_a \\ x_b & y_b & z_b \end{vmatrix} = 0$$

평면 EAB는 상수 c_1, c_2, c_3 를 사용하여 식(4-1)과 같이 나타낼 수 있다.

$$\begin{aligned} c_1x + c_2y + c_3z &= 0 & (4-1) \\ c_1 &= y_a z_b - y_b z_a \\ c_2 &= x_b z_a - x_a z_b \\ c_3 &= x_a y_b - x_b y_a \end{aligned}$$

선분 AB와 선분 PQ가 서로 만나기 위해서는 식(4-2)에서 구한 λ_1 이 0과 1 사이에 있으면 된다.

$$\lambda_1 = \frac{c_1 x_p + c_2 y_p + c_3 z_p}{c_1(x_q - x_p) + c_2(y_q - y_p) + c_3(z_q - z_p)} \quad (4-2)$$

식(4-2)로부터 구한 λ_1 을 식(4-3)에 대입함으로써 물체의 높이 및 위치를 알 수 있다.

$$\vec{EP} + \lambda_1 \vec{PQ} = (x_p, y_p, z_p) + \lambda_1(x_q - x_p, y_q - y_p, z_q - z_p) \quad (4-3)$$

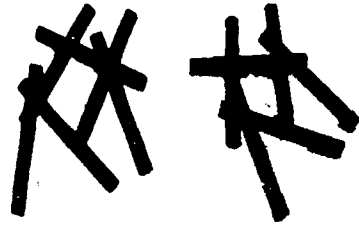
5) 실험 결과

본 연구에서의 vision system은 CCD(Coupled Charged Device)카메라 2대, 512x512의 해상도와 256 gray level를 갖는 영상카드, 5축로봇 1대, PC 1대로서 구성되어 있으며, 좋은 영상을 잡기 위하여, 하얀 백지 위에 분필은 검정색을 칠해서 사용하였다.

본 논문에서는 binary를 사용하여 image processing을 시키고, 오목점 및 볼록점, CAD data, distance, slope를 이용하여 image segmenting을 시켰으며, weight 계산 후 optimality test에 의해서 image modeling을 하였으며, 양쪽 카메라의 corresponding objects를 찾기 위하여, bipartite matching 방법을 사용하였다. 물체의 양쪽 방향 중에서 최소한 한쪽 방향에서 물체의 특징점을 추출할 수 있도록 물체를 놓았다. 그림 6은 양쪽 카메라에서 잡은 영상을 나타낸다. 그림 7은 image processing

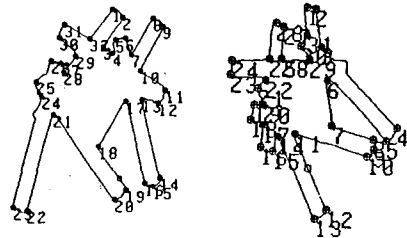
후에 쌓여 있는 물체들의 꼭지점을 나타낸 것인데 진한 검정색은 오목점(concave point)이고, 나머지는 볼록점(convex point)을 나타낸다.

그림 8은 카메라 1, 2의 image segmenting 과정을 보여 주고 있으며, 그림 9는 optimality test후의 서로 분리된 물체들을 보여 주고 있으며, 그림 10은 물체들의 기하학적인 관계를 나타낸 것으로서 큰 글자는 물체가 쌓여 있는 순서를 나타낸다. 그림 11은 로봇트가 제일 위에 있는 물체를 들어 올리는 모습이고 그림 12는 제일 위에 있는 물체부터 차례로 작업한 후 제일 아래에 있는 물체를 들어 올리는 모습이다.



camera-1 camera-2

그림 6 화면상에 나타난 이미지



camera-1 camera-2

그림 7 이미지 processing후의 물체들의 꼭지점

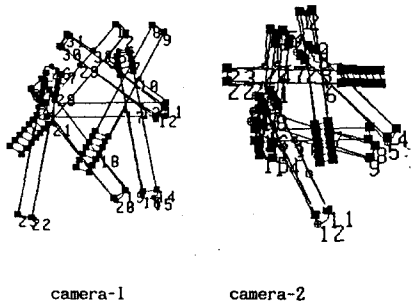


그림 8 segmenting process의 보기

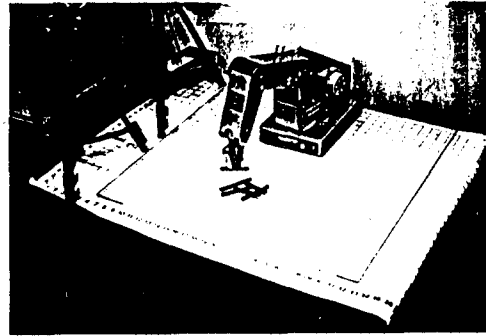


그림 11 로봇이 제일 위에 있는 물체를 집는 모습

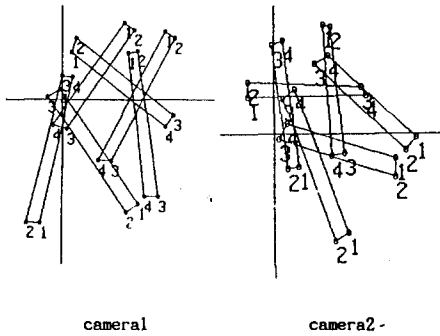


그림 9 optimality test후 분리된 물체들

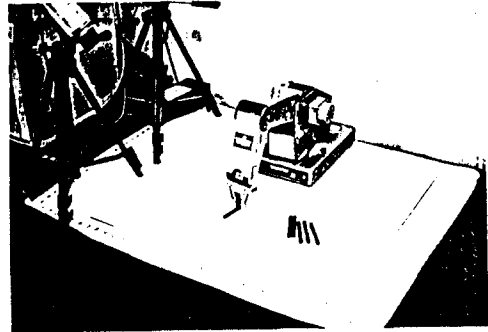


그림 12 로봇이 순서적으로 작업한후 마지막으로 제일 밑에 있는 물체를 잡는 모습

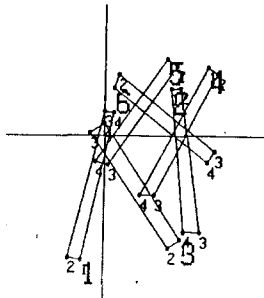


그림 10 쌓여있는 물체들의 기하학적인 관계

6) 결론

본 연구에서는 카메라 두 대를 사용하고, binary를 이용하여 image processing을 시키고, 서로 대칭이고, 특징 점을 찾기 힘든 경우에 image segmenting을 위하여 CAD정보, concave point property, orientation, distance 정보를 사용했으며, image modeling을 위하여, weight를 계산한후 weight가 0.25, 0.5, 0.75인 경우에 optimality test를 하였다. 그리고 각 물체의 높이를 구하기 위해서, camera-1, camera-2의 corresponding object 및 vertex를 식별하기 위하여 bipartite matching method를 사용하였

다.

두 물체가 서로 교차하는 경우, 위 아래의 물체를 구별하기 위하여, 두물체의 교차점을 구한후 교차점에서의 각 물체의 3D정보를 구한 다음, 제일위에 있는 물체부터 순서적으로 위치 및 방향정보를 robot에 제공함으로써, 제일위에 있는 물체부터 Robot가 작업을 하게된다.

물체가 많이 쌓여 있을 때, 노이즈, 외곽현상 등으로, 카메라만으로 인식하는 데에는 한계가 있기 때문에, stripe line raser를 사용 함으로써, 좀더 정확한 위치 및 방향을 구할 수 있을 것이다.

참 고 문 헌

- [1] Min_Hong Han and Sangyong Rhee " Camera calibration for 3_D measurement", Pattern Recognition, Vol 25,NO.2, pp. 154 - 164. 1992
- [2] Kelly R., Birk J., Martins H. and Tella R.(1982):Robot System Which Acquires Cylindrical Workpieces from Bins",IEEE Trans.on Systems,Man and Cybernetics,Vol. SMC-12, no 2, March/April, pp. 204-213.
- [3] McEntire R.(1976):" Three Dimensional Accuracy Measurement Methods for Robots",The Industrial Robot,Vol 3, no. 3, September, pp 105-112.
- [4] 신영수, 김현석 " 3차원 그래픽", 가남사.
- [5] Robert M. Haralick and Linda G. Shapiro " Computer and Robot Vision" Vol. I.
- [6] Robert M. Haralick and Linda G. Shapiro " Computer and Robot Vision" Vol. II.
- [7] Dana H. Ballard and Christopher M. Brown " Computer Vision" Prentice Hall.
- [8] Horn B.(1986):"Robot Vision", John Wiley & Sons, New York.
- [9] Robert J. Schilling:"Fundamentals of Robotics", Prentice- Hall
- [10] Crag "Introduction of Robots" ,Addison Wesley