

시각 센서가 있는 로봇 제어의 컴퓨터 그래픽
시뮬레이션에 관한 연구

박 남 룰, 김 경 기
한양대학교 전자공학과

A study on the computer graphic simulation of a robot control
with a visual sensor.

Nam-Ryul Park Kyung-Ki Kim
Han-Yang University

Abstract

In this paper, a new algorithm is proposed to obtain important corner and real corner points of the simple objects and the visual robot graphics simulation using the results is studied. This is capable of performing as a debugging tool for task programming of a visual robot. And the robot motion is also simulated on a CRT terminal.

I. 서 론

시각 능력은 로봇이 복잡한 환경에서 지적이고 융통성 있게 움직이게 하는 능력을 부여해 줄 수 있다. 이러한 로봇 미션에서는 조명의 구조적 위치, 사진 처리 작업, 효과적인 정보 추출 방법 등이 충분히 고려되어야 한다. 잘못된 물체 인식 정보에 의한 로봇 제어시 물체의 파손, 로봇의 손상, 부정확한 로봇의 동작을 유발할 수 있다. 이를 사전에 방지하고 보완하기 위해서는 디버깅 도구로서 시각 로봇 그래픽 시뮬레이션 장치가 요구된다. 이를 위해서 본 논문에서는 평면상에 놓여 있는 단순한 물체를 CCD 카메라를 사용해 인식하여 사진 처리 작업을 한 후 물체의 특징을 잘 나타내는 중요 모서리 점들과 실 모서리 점들을 발견하는 방법을 제안하였으며 또한 그 결과를 이용하여 자체적으로 모니터상에 물체와 로봇을 표시하여 로봇 동작을 디버깅하는 시각 로봇 그래픽 시뮬레이션에 관해 연구하였다.

II. 회상 사진 처리 작업.

2.1 스무딩 작업. (Smoothing)

처음 입력 회상을 샘플링(sampling), 양자화, 전송 또는 외판으로 인해 잡음이 존재한다. 이 잡음과 물체 윤곽선의 불명명을 축소하기 위해서 매니언 필터를 사용한다. 이것은 가로, 세로가 각각 3 인 원도우의 최소(pixel)값을 소정(sorting)한 후 좌, 우 최소값들의 배열이 대칭이 되도록 하는 매니언 값을 선택하여 중심 최소에 할당한다. 이 회상은 전체 회상(256 * 256)에서 수평, 수직으로 이동하면서 반복하여 처리한다.

2.2 물체 윤곽 강조(Edge enhancement)

물체의 바깥의 경계선 부분을 강조하기 위해서 수평 수직으로 최소값의 변화가 큰 부분을 국부적(local) 미분 연산자인 sobel 연산자를 사용해 검출해 준다.

a	b	c
d	p	e
f	g	h

$$S_i = |S_i^x| + |S_i^y|$$

$$S = [S_i^x + S_i^y]^{\frac{1}{2}}$$

그림 1. 일반적 3*3 원도우

$$S[i,j] = [(c + 2e + b) - (a + 2d + f)] + [(f + 2g + h) - (a + 2b + c)] \quad (2)$$

여기서는 256 개의 명암도(light intensity level)내에 S 값이 포함되도록 하기 위해서 * S 값을 취하였다.

2.3 이진 회상 획득(Thresholding)

이진 회상을 획득하기 위해서 대역적(global) threshold 값을 취하는 것 대신에 국부적(local) threshold 값을 이용하고 그 값에 임계값 T_L 를 적용한다. 3 * 3 원도우에 대한 Local threshold 식은 아래와 같다.

$$Tf(i,j) = \sum_{p,q} W_{pq} * f(i+p, j+q) \quad (3)$$

W_{pq} : 미리 지정된 가중치
 W_{00} : 다른 가중치 합이 음수

$$x(x,y) = \begin{cases} 1 & \text{if } |Tf(x,y)| > T_L \\ 0 & \text{if } |Tf(x,y)| < T_L \end{cases} \quad (4)$$

여기서 $T_L = 450$ 이다.

III. 이진 회상 처리 작업

3.1 디지털 스무딩(Digital smoothing)

이진 회상이 불규칙한 윤곽, 물체내의 작은 구멍들(1 개의 pixel), 물체 구성들의 두께, 분리된 점들과 같은 잡음 현상이 발생하는 경우에 스무딩 작업이 필요하다.

$$\begin{aligned}
 D1 &= p + b * g * (d + e) + d * e * (b + g) \\
 D2 &= p * [(a + b + d) * (e + g + h) + (b + c + e) * \\
 &\quad (d + f + g)] \\
 D3 &= \overline{p} * (d * f * g) * \overline{(a + b + c + e + h)} + p \quad (5) \\
 D4 &= \overline{p} * (a * b * d) * \overline{(c + e + f + g + h)} + p \\
 D5 &= \overline{p} * (e * g * h) * \overline{(a + b + c + d + f)} + p \\
 D6 &= \overline{p} * (b * e * e) * \overline{(a + b + f + g + h)} + p
 \end{aligned}$$

D1 : 물체 내의 구멍(1개 pixel) 및 V 가 부분(notch) 부분.

D2 : 분리된 점들을 제거.

D3,D4,D5,D6 : 물체 구성 단위점 부분.

* : 논리곱 + : 논리합 - : 논리역
 (* 그림1 참조)

$$P = \begin{cases} 1 & \text{if } Dn = 1 \quad , n=1,2,\dots,6 \\ 0 & \text{if } Dn \neq 1 \end{cases} \quad (6)$$

3.2 물체 윤곽선 추적(Tracking outlines)

전체 이진 회상에서 각 최소값은 그것이 물체 윤곽선 상에 놓여 있으면 1, 그렇지 않으면 0 값을 갖는다. 그러므로 전체 회상을 메모리에 저장하려 한다면

메모리가 낭비되고 처리 시간이 길다. 회상에서 필요한 것은 단순히 물체의 윤곽선들이므로 물체와 배경의 경계선을 추적하여 물체의 윤곽을 알아내는 사슬 부호(chain code) 방법을 사용한다. 먼저 Forward 추적에 대해 설명하면 물체의 정확한 윤곽점인 초기점을 찾기위해 회상을 raster scan 한다. 그 점을 발견하면 그 위치에서 그림2의 사슬 부호를 이용하여 시계방향으로 이웃점을 찾는다. 이웃점의 선택 조건은 값이 1 인 어딘 방향 점들 중 반 시계 방향으로 한개 이전 점이 0 이어야 한다. 그 다음 이웃점은 바로 전의 이웃점 방향 바로 다음 부호에서 부터 시계 방향으로 앞에서 설명한 이웃점 조건을 만족하는 점을 선택한다.

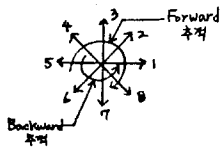


그림 2. 사슬 부호 분기점으로 되돌아 올때까지 backward 추적한다. 분기점으로 되돌아오면 다시 forward 추적 과정이 반복된다. 위와 같은 과정에 의해서 초기점으로 되돌아오면 윤곽선 추적을 끝내고 다른 물체를 추적시 중복을 피하기 위하여 이미 추적한 물체를 회상에서 지운다. 다시 다음 물체를 찾기위해 전 물체의 초기점에서 부터 raster scan 한다.

IV. 물체 중요 코너 점 및 실 코너 점 발견

물체를 알기 위해서는 여러가지 요소가 있을 수 있지만 그것의 특징을 간단하고 쉽게 나타내어 주는 것은 물체의 코너 부분들이라 할 수 있다. 물체의 중요 코너 점들을 발견하여 그 코너 점들을 순서적으로 연결하면 복잡한 물체를 단순화시킬 수 있으며 그 물체에 대한 정보량도 축소시킬 수 있다. 그리하여 본 논문에서는 그래픽 시뮬레이션을 위한 단순하고 빠른 중요 코너 점 및 실 코너 점 발견 방법을 제안하였다. 물체의 코너 점은 같은 방향을 갖는 연속적인 점들에서 다른 방향을 갖는 연속적인 점으로 변화하는 부분에서 발생한다. 그러므로 이러한 코너 특성을 갖는 점들을 모두 발견한다. 중요 코너 점 및 실 코너 점들도 이러한 점들에서 발생하므로 위에서 발견한 코너 점들에서 아래의 방법을 적용한다.

먼저 어떤 코너 점 C_n 에서 $n-k$ 번째 점 C_{n-k} 의 사슬 부호 값을 $An-k$ 이라 할 때 C_{n-k} 이전 점들 중에 $An-k$ 와 같은 사슬 부호 값을 갖는 연속적인 점의 갯수를 $Bp(k)$ 라 하고 $n+k$ 번째 점 C_{n+k} 의 사슬 부호 값을 $An+k$ 이라 할때 C_{n+k} 이후 점들 중에 사슬 부호 값 $An+k$ 과 같은 사슬 부호 값을 갖는 연속적인 점의 갯수를 $Fp(k)$ 라 한다.

다음으로 코너 점 C_n 에서 1 번째 이전 코너 점 C_{n-1} 까지의 벡터 $Bv(1)$ 과 C_n 에서 1 번째 이후 코너점 C_{n+1} 까지의 벡터 $Fv(1)$ 을 구한다. 이전 벡터 $Bv(1)$ 을 기준 벡터로 하여 두 벡터 $Bv(1)$ 과 $Fv(1)$ 의 내적(inner product) $TH(1)$ 을 구한다. 구한 값을 아래 식에 대입한다.

$$COR = \frac{\sum_{i=1}^{cn} Bp(i) * Fp(cn-i+1)}{[\sum_{j=1}^{cn} TH(j)] * (1/cn)} \quad (7)$$

$$0 < Bp(k) \leq Pmax$$

$$0 < Fp(k) \leq Pmax$$

cn : 주어진 입력값

$Pmax$: 지정되는 최대점 갯수의 한계값

$Bp(k)$ 또는 $Fp(k)$ 의 값이 크고 $TH(1)$ 값이 작을수록 코너 점의 특징은 현저하므로 전체 코너 점의 COR 값에 대한 임계값 $Tcor$ 에 의해 중요 코너 점들을 발견한다.

위에서 결정된 중요 코너 점들 중에서 2 개 또는 3 개 중요 코너 점들이 서로 이웃해 있거나 한점 차이로 이웃해 있는 경우, 양쪽 점의 전 또는 후 방향의 사슬 부호가 연속적인 홀수 부호 값을 갖으면 양 극단점 각각에서 그 전 또는 후 중요 코너 점과의 벡터들의 교차점을 그 코너의 실 코너 점으로 본다.

V. 물체 인식 과정

물체들을 인식하기 위해서는 물체 서로간의 상관 관계, 물체의 면적, 무게 중심, 물체의 주축, 물체 폭 과 너비 등의 요소들을 구하여야 한다.

여기서 물체의 방향은 주축 방향으로 하고 물체의 중심점은 무게 중심점으로 한다. 위의 각 요소 값들의 계산은 물체에 대한 사전 지식으로 일괄적인 높이값을 사용한다. 한 물체 내에 포함된 다른 물체의 식별은 내부에 있는 물체의 각 중요 코너 점들이 외부 물체의 각 중요 코너 점 내부에 있으면 내부에 있는 것으로 본다. 또한 내부에 물체를 포함하고 있는 물체의 무게 중심은 외부 물체의 무게 중심을 전체 무게 중심으로 취급한다.

VI. 시스템 좌표계

6.1 카메라 좌표계

카메라는 물체 공간 상에 있는 점들을 카메라 화상에 일치하는 점들로 나타낸다. 그러므로 토포트 비전을 위해서는 카메라 화상에 있는 점들의 좌표값을 이용해 실제 물체 공간 좌표상의 점 위치를 알아내야 한다. 이러한 과정은 원근(perspective) 변환 관계를 구함으로써 해결될 수 있다.

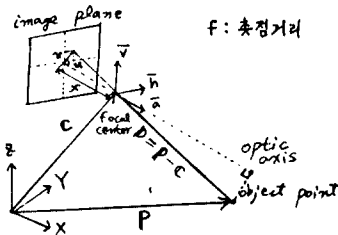


그림 3. 카메라 좌표계의 파라미터들.

위의 그림에서 카메라의 방향은 영상면에 수직인 단위 벡터 \vec{h} 로 표시되고 단위 벡터 $\vec{h}, \vec{v}, \vec{a}$ 등은 서로 수직이다.

그림 3에서 삼각형들을 서로 비교한 결과는

$$\frac{u}{f} = \frac{D \cdot \vec{h}}{D \cdot \vec{a}}, \quad \frac{v}{f} = \frac{D \cdot \vec{v}}{D \cdot \vec{a}} \quad (8)$$

이다.

여기서 벡터 \vec{D} 는 초점 중심에서 물체 점까지의 벡터이며 $\vec{D} = \vec{P} - \vec{C}$ 이다.

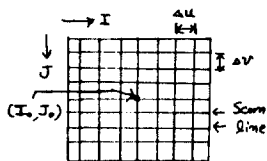


그림 4. 샘플링된 화상.

아날로그 좌표값 u, v 를 사용하는 것 대신에 그림 4에서의 I, J 를 사용하면 u, v 는

$$u = (I - I_0) \Delta u, \quad v = (J - J_0) \Delta v \quad (9)$$

$\Delta u, \Delta v$ 는 the sampling interval 이다.

식(9)를 식(8)에 대입하면 결과적인 방정식은

$$I = \frac{D \cdot \vec{h}}{D \cdot \vec{a}}, \quad J = \frac{D \cdot \vec{v}}{D \cdot \vec{a}} \quad (10)$$

$$\vec{H} = \frac{f}{\Delta u} \vec{h} + I_0 \vec{a}, \quad \vec{V} = \frac{f}{\Delta v} \vec{v} + J_0 \vec{a}$$

이다.

여기서 \vec{H} 와 \vec{V} 는 단위 벡터들이 아니며 단위 벡터 \vec{a} 와 수직이 아니다. 위 방정식들을 풀어 벡터 \vec{D} 를 구하면 기준 좌표계에 대한 물체 점의 위치를 알 수 있다.

6.2 토포트 좌표계

물체를 이용하는 것과 같은 토포트 제어시 R_T 를 구하여 Inverse Kinematics를 풀어야 한다.

그러므로 기준 좌표계 W 에서 물체외 토포트 좌표계를 이용하여 R_T 를 구하면

$$\begin{aligned} {}^W T &= {}^W T^R T E \\ {}^R T &= {}^W T^R T E^{-1} \end{aligned} \quad (11)$$

이다.

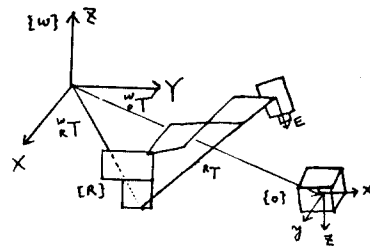


그림 5. 토포트 좌표계.

VII. 실험 및 고찰.

CCD 카메라를 토포트의 work space 내에 설치하여 original image를 획득한 후 IBM PC 모니터 상에 나타나는 실험에 따라 컴퓨터와 인간이 대화하는 형식으로 프로그램을 수행하도록 하였으며 디버깅시 도움이 되도록 하였다. original image를 처리한

edge detection 한 결과를 본 논문에 제시한 중요 코너 점 및 실 코너 점 방법으로 처리하여 중심 코너 점 및 실 코너 점을 발견하였다. edge detection 결과는 조명의 영향에 따라 민감하게 좌우되므로 세심한 주의를 기울여 실험하였다. 카메라 화상면 상의 각 점을 십자 좌표계의 각 점으로 변환하여 물체 인식 과정에서 필요한 요소를 얻어내었다. 그래픽 시뮬레이션을 위해 각 점을 IBM PC 모니터 상에 표시하여 연결하여 주었고 물체의 중심점에 대한 Inverse kinematics 를 풀어 물체를 Pick up 하는 작업을 실행하였다.

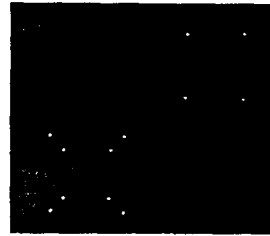


그림 8. 중요 코너 및 실 코너 발견 결과.

VIII. 결 론

본 논문에서는 CCD 카메라를 사용한 로봇의 비전 그래픽 시뮬레이션에 관해 연구하였다. 이러한 그래픽 시뮬레이션을 위해서 본 논문에 제시한 중요 코너 발견 및 실 코너 방법이 물체의 많은 점들을 모두 기억하는 메모리 상의 문제와 복잡성을 단순화시킬 수 있음을 확인할 수 있었고 그래픽 시뮬레이션에 잘 적용됨을 알 수 있었다. 그러나 좀 더 나은 시뮬레이션 장치가 되기 위해서는 복잡한 물체에 대한 처리 방법의 연구가 더 필요하며 그 외, 컴퓨터 처리 용량과 속도, 주변 장치의 하드웨어화, 조명의 극복 등의 연구가 요구된다.

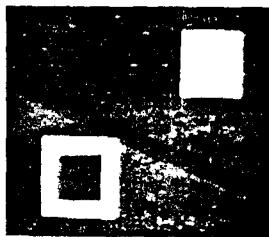


그림 6. original image.

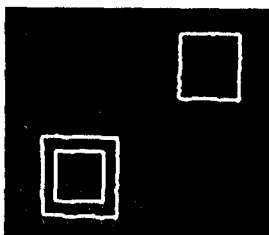


그림 7. edge detection 한 결과.

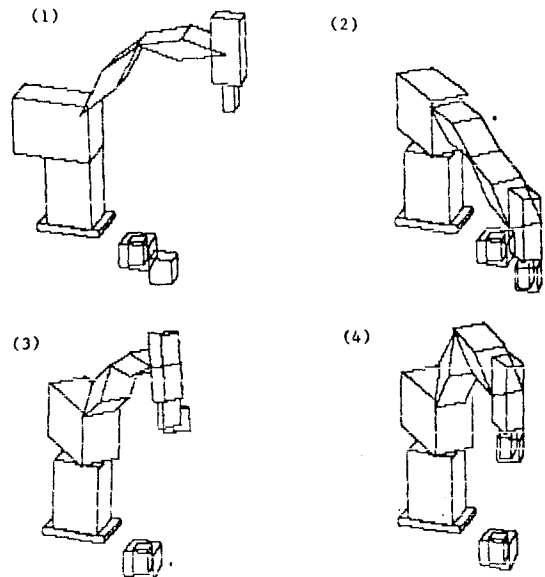


그림 9. 로봇 그래픽 시뮬레이션 결과.

IX. 참고 문헌

- [1] H.Freeman, " Computer processing of line drawing images", Comput. Surv. Vol.6, NO.1, March, 1974
- [2] T.H.Morrin, "Chain-link compression of arbitrary black-white images", Computer Graphics and Image Processing, 5, pp.172-189, 1976
- [3] H.Freeman and L.S.Davis, " A corner finding algorithm for chain-coded curves.", IEEE Trans. Computer, vol.C-25, pp 197-203, Mar., 1977
- [4] A.Rosenfeld and E.Johnston, " Angle detection on digital curves", IEEE Trans. Comput. vol.C-22, Sept. 1973
- [5] R.C.Gonzalez and P.Wintz, Digital image processing, Wesley, 1987
- [6] T.Sata, F.kimura and A.Anano, "Robot Simulation system as a task programming tool", The 11th international symposium on industrial Robots, 1981