

힘제어 기반의 틈새 추종 로봇의 제작 및 제어에 관한 연구 : Part I. 신경회로망을 이용한 레이저와 카메라에 의한 틈새 검출 및 로봇 제작

Implementation and Control of Crack Tracking Robot Using Force Control : Crack Detection by Laser and Camera Sensor Using Neural Network

조 현 택, 정 슬*
(Hyun Taek Cho and Seul Jung)

Abstract : This paper presents the implementation of a crack tracking mobile robot. The crack tracking robot is built for tracking cracks on the pavement. To track cracks, crack must be detected by laser and camera sensors. Laser sensor projects laser on the pavement to detect the discontinuity on the surface and the camera captures the image to find the crack position. Then the robot is commanded to follow the crack. To detect crack position correctly, neural network is used to minimize the positional errors of the captured crack position obtained by transformation from 2 dimensional images to 3 dimensional images.

Keywords : crack sealing robot, crack detection algorithm, neural network

I. 소개

도로 포장의 구조는 표층과 노반으로 되어 있다. 표층은 교통 하중을 하부의 노반이나 노상에 전달함과 동시에 마모에 견디고 하중에 대한 저항이 비교적 크며 방수성을 갖추도록 설계가 되어 있다. 하지만 도로 위를 지나는 자동차의 교통 하중에 의해서 시간이 지남에 따라 점차 도로의 표층면에 결함이 생긴다. 그리고 결함으로 생긴 틈새로 물이 노반으로 들어가면 노반을 약하게 만들어 많은 자본을 들여서 만든 도로가 오랜 수명을 가질 수 없게 되므로 계속적인 유지 및 보수하는 작업이 필요하게 된다.

그 중에서 도로의 틈새를 봉합하는 일은 매우 지루하고 반복적이며 작업 속도가 느린 작업이다. 특히 위험한 것은 움직이는 자동차에 의해서 작업자가 사고가 생길 수 있고 봉합 품질의 균일성이 작업자의 기능에 따라 차이를 보일 수도 있다. 이에 품질과 생산성의 향상 및 작업자의 위험요소 제거를 위해 도로 틈새를 봉합하는 자동화에 대한 연구가 요구된다. 선진국에서는 오래 전부터 도로의 유지 보수를 위해 많은 투자를 해 왔으며 현재 몇몇 제품은 상용화되고 있는 실정이다.

특히 미국 캘리포니아 주의 도로 교통국(CALTRAN)에서는 UC DAVIS대학에 AHMCT(Advanced Highway Maintenance & Construction Technology)란 센터를 세우고 도로의 유지 및 보수에 관한 다양한 연구를 하고 있다. 연구의 내

용들을 살펴보면 도색로봇, stenciling robot, 도로 보수 로봇 등 다양한 첨단화 된 로봇을 연구하고 있다. 도로의 틈새를 자동으로 봉합하는 시스템에 대한 연구에 대해서 카메라와 이동로봇의 위치 추정 방법으로 틈새를 찾고 추종하며 이동로봇에 있는 로봇팔로 봉합하는 실험을 하였다[1]. 국내에서는 이러한 분야의 연구가 아직 활발하지 못하며 비슷한 응용으로 터치센서를 사용하여 이동로봇이 격자 형태의 용접 틈새를 인식하고 추종하면서 용접하는 연구를 수행하였다[2].

카메라와 레이저를 사용하여 영상을 획득하는 방법은 활발하게 연구되어 왔다. 3차원 측정 정보를 얻기 위하여 두 개의 라인 스캔 카메라를 중심축으로 360° 회전하여 사방의 3차원 정보를 획득하였고 복잡한 매니플레이터와 운반체의 안내와 경로 추적에 적용하였다[3]. 레이저 스캐닝 검사를

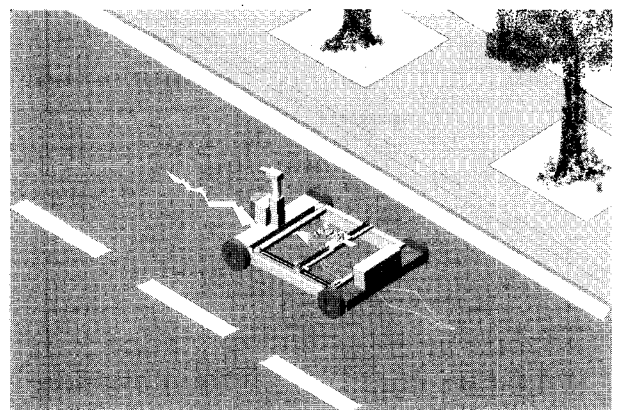


그림 1. 틈새 로봇의 가상 작업 환경.

Fig. 1. Virtual working environment of the robot.

* 책임저자(Corresponding Author)

논문접수 : 2004. 1. 2., 채택확정 : 2004. 10. 30.

조현택, 정슬 : 충남대학교 메카트로닉스공학과

(hyuntkc@empal.com/jungs@cnu.ac.kr)

※ 본 논문은 2002년도 학술진흥재단 지방대 육성지원 KRF-2002-02-D00076에서 지원하여 연구하였음.

통하여 결합 측정과 결합 크기의 측정기준 방법을 제시하였다[4]. 레이저 슬릿빔과 두 대의 카메라의 관계를 이용한 삼각방법을 제시하였고 사람의 얼굴에 대하여 3차원 정보를 획득하고 구축하는 방법을 수행하였다[5]. 기존의 스테레오 매칭 방법에 대한 복잡한 반복 과정을 거치게 되는 문제점을 극복하기 위하여 스테레오의 두 영상 면의 점을 새롭게 매칭시키는 MRF(Markov random field)방법을 제시하였다[6]. 운반체에 장착된 두 대의 레이저 슬릿빔과 단일 카메라 및 삼각 기법을 이용하여 물 속 환경의 용접 결합이나 용접된 접합부의 각지 실험을 수행하였다[7]. 3차원 비전 검사 시스템에 대한 물체 표면 반사광을 피하는 방법으로 각각의 단일 레이저 슬릿빔과 CCD 카메라를 사용하였으며, 삼각 알고리즘과 필요한 정보에 인접한 불필요한 3차원 데이터 점의 군집 점들을 제거하는 연구를 수행하였다[8]. 물 속 작업 환경에서 구조화된 라인 레이저와 카메라를 가지고 물체를 스캔하고 삼각 방정식과 자동 문턱 값 조절을 이용하여 3차원 CAD를 구현하였으며 물의 탁도, 물체의 각도 및 거리에 따라 물체의 크기를 측정하였다[9].

이동로봇의 구동방법으로는 신경망을 이용하거나 비선형 제어인 백스테핑 제어기를 사용하여 이동 로봇의 추종 문제와 제어 문제를 동시에 만족하는 실험을 수행하였다[10-12]. 본 연구의 목적은 틈새의 형상을 인식하고 틈새경로에 대한 정보를 획득하여 이동로봇이 획득한 틈새를 따라가면서 틈새를 밀봉하는 시스템을 개발하는 것이다. 이러한 자동화 시스템의 개발을 위해 필요한 틈새의 인식과 이동로봇의 틈새 추적에 관한 연구에 중점을 두어 연구하였다. 그림 1은 가상적인 환경을 도식적으로 보여준다.

II. 틈새 추종 로봇 시스템 구성

1. 전체 시스템 개요

틈새 추종 로봇 시스템은 크게 세 부분으로 나뉘어진다. 첫 번째는 틈새를 인식하는 센서 시스템이고 두 번째는 측정된 틈새를 따라가는 이동로봇과 XYZ 축으로 선형 조인트를 가지는 로봇이고 세 번째는 틈새 인식 센서와 로봇을 움직이는 제어기로 나눌 수 있다. 실제 제작한 틈새 추종 로봇 시스템은 그림 2와 같다. 검은 케이스로 덮여있는 곳이 틈새를 인식하는 센서 부분이고 가운데 부분은 XYZ축 직교 로봇이며, 뒷 부분이 센서와 로봇을 제어하기 위한 컴퓨터와 DSP 시스템이 있는 부분이다. XYZ 직교 로봇의 목적은 노면과의 접촉을 통해 틈새의 이물질 제거하는 역할과 함께 틈새를 메우는 작업을 하게 된다. 이를 위해 XYZ 직교 로봇의 z 축에는 힘센서가 부착되어 있으며, z축의 움직임은 볼 스크루 방식으로 만들어 졌다. 또한 자율 주행을 위한 배터리가 내장되어 있다. 전체 시스템의 길이는 1.6m이고 폭은 0.8m이며 전체 무게는 약 100kg정도이다. 전체적인 구조에 대한 블록 다이어그램은 그림 3과 같다. 컴퓨터는 카메라의 영상을 획득하고 분석하여 틈새를 인식하여 DSP 시스템으로 틈새 데이터를 넘겨준다. DSP 시스템은 이러한 틈새 데이터를 가지고 이동 로봇의 좌우 바퀴를 구동하여 이동 로봇이 틈새를 추적하게 하며 또한 X-Y-Z축 테이블을 움직인다.

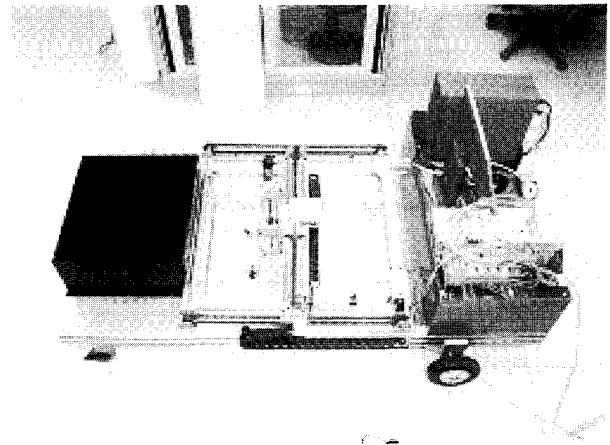


그림 2. 실제 시스템의 측면. Fig. 2. Real robot.

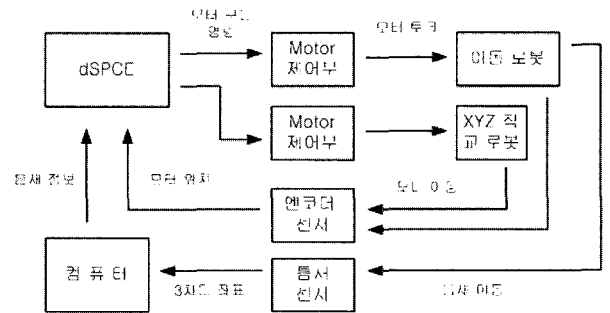


그림 3. 틈새 추종 로봇 시스템 구조 블록. Fig. 3. Block diagram of the robot.

2. 틈새 센서

그림 4는 전체 센서 시스템의 구성 및 좌표계에 대한 설명이다. 먼저 레이저 다이오드에서 나오는 레이저 슬릿광의 좌표계[x,y]와 이미지 평면의 좌표계[u,v]로 이루어져 있다. 그림 5에서 알 수 있듯이 레이저 슬릿광을 카메라로 찍으면 y축 좌표가 이미지 평면에서 v축 좌표로 변환된다. 이를 바탕으로 레이저 슬릿광 평면에 대한 3차원 데이터로 변환시킬 수 있다. 그림 5는 실제 제작한 센서 시스템으로 카메라와 레이저 슬릿광의 광원과의 거리는 260mm이고 레이저 슬릿광과 바닥과의 거리도 260mm이다. 따라서 레이저 슬릿광과 이미지가 이루는 각도는 45°가 된다. 레이저 다이오드는 Stocker Yale사의 SNF 타입으로 660nm의 레이저 파장, 35mW 파워 및 60°의 팬각도를 가지고 있다. 중요한 특징으로는 길이에 따른 선의 형태가 선형성을 나타낸다. 카메라는 산업용으로 나온 CCD 카메라로 640×480의 24bit 칼라 이미지 가지고 있으며, 컴퓨터에 있는 Data Translation사의 DT3132 캡처보드에 의해서 초당 30 프레임으로 이미지를 획득한다.

3. 갠트리 형태의 로봇

센서에 의해 감지된 노면의 틈새는 xyz 방향으로 움직이는 갠트리 형태의 로봇에 의해 깨끗하게 청소된다. 틈새를 밀봉하기 전에 먼저 틈새에 낀 이물질을 제거하는 작업을

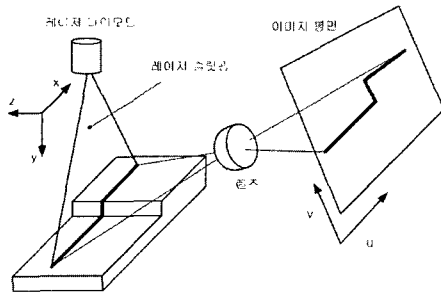


그림 4. 센서 시스템 구성.
Fig. 4. Sensor configuration.

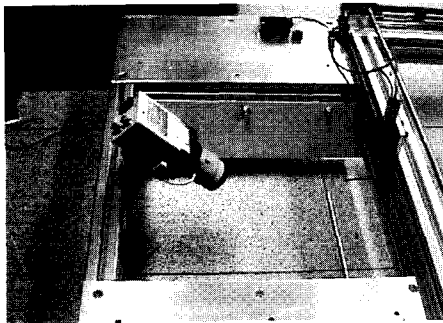


그림 5. 제작한 센서 시스템 측면.
Fig. 5. Real sensor environment.

로봇이 하게 된다. 이 청소작업을 로봇이 효율적으로 하기 위해서 노면과의 접촉을 일정한 힘으로 제어하도록 되어있다. 그림 2에서 로봇의 z축 맨 끝에 힘센서가 부착되어있는 것을 볼 수 있다. 능동적인 힘제어를 사용하여 노면과의 적당한 접촉을 유지하며 작업을 효율적으로 실행하고자 하는 것이다. 힘제어 부분에 대한 자세한 내용은 다른 논문에서 다루고 있다.

4. 전원부

틈새 추종 로봇이 자율주행을 하기 위해서는 독립전원을 사용하여야 한다. 여기에서는 자동차용 납축전지와 700W 인버터를 이용하였다. 인버터는 직류 소스를 교류 소스로 변환시켜주는 장치로 제어에 필요한 컴퓨터를 구동시켜주는 장치이다. 그리고 제어회로에 필요한 5V와 12V는 인버터에서 나오는 교류 전원을 다시 SMPS를 이용하여 직류 전원으로 변환하여 사용하였다. 또한 모터 쪽의 전기신호가 제어 회로 쪽에 영향을 주지 않기 위해서 각각 다른 납축전지를 사용하였다. 모터에 사용한 납축전지는 12V 40AH 용량의 납축전지 2개를 직렬로 연결하여 24V인 150W와 70W 모터 4개를 구동하고 인버터 쪽에는 12V 80AH 납축전지 하나를 사용하였다.

III. 틈새 추출

1 틈새의 특징점 추출

1.1 틈새의 특징점 추출

틈새의 모양은 여러 가지 형태로 나타날 수 있으나 여기에서는 틈새의 모양 중에 그림 6처럼 가장 간단한 형태의 틈새에 관해서 연구하였다.

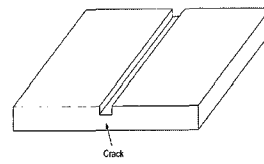


그림 6. 크랙의 형태.
Fig. 6. Shape of crack.

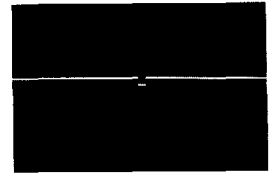


그림 7. 정면 틈새.
Fig. 7. Center crack.

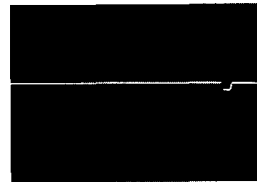


그림 8. 우측 틈새.
Fig. 8. Right crack.

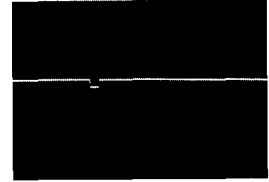


그림 9. 좌측 틈새.
Fig. 9. Left crack.

그림 7, 8, 9는 레이저 슬릿광을 전처리 과정만 수행한 영상으로 그림 7은 틈새가 정면에 있을 때이고 그림 8은 오른쪽에, 그림 9는 왼쪽에 있는 경우이다. 그림 7, 8, 9에서 공통적으로 나타나는 이미지의 특징은 여러 개의 직선으로 이루어져 있으며, 특히 틈새의 한 쪽은 직선이 끊어져 있는 것을 알 수 있다. 이것을 이용하여 슬릿광을 클러스터링 처리를 하면 그림 7에서는 3개, 그림 8과 9에서는 2개의 클러스터링으로 나누어진다. 이렇게 나누어진 클러스터링의 끝점을 이용하여 틈새의 시작점과 폭을 계산할 수 있다.

2. 3차원 영상정보의 추출

그림 4의 센서 시스템을 이미지 좌표계로 나타내면 그림 10과 같이 나타낼 수 있다. world plane은 레이저 슬릿광의 평면이고 CCD plane은 카메라의 CCD 평면이며 frame buffer는 실제 캡처된 영상을 나타내고 있다. 이때 카메라의 모델을 이상적인 핀 홀 모델이라고 가정하면 전역 좌표계(슬릿광 평면)에 나타난 $P(X, Y)$ 는 카메라의 CCD 센서에 $\bar{p}(x, y)$ 로 나타내야 하지만 카메라 렌즈의 왜곡에 의해서 $p(x, y)$ 에 나타나고 영상에는 $p(u, v)$ 로 표시된다.

여기에서 렌즈에 의한 radial, decentering, thin prism 왜곡을 고려하면 (1)과 (2)로 설정할 수 있다[13].

$$x = (u - u_o)d_x, y = (v - v_o)d_y \tag{1}$$

$$\bar{x} = x + \delta^{(x)}(x, y), \bar{y} = y + \delta^{(y)}(x, y) \tag{2}$$

$$\begin{aligned} \delta^{(x)}(x, y) &= k_1x(x^2 + y^2) + (p_1(3x^2 + y^2) + 2p_2xy) \\ &\quad + s_1(x^2 + y^2) \\ \delta^{(y)}(x, y) &= k_1y(x^2 + y^2) + (2p_1xy + p_2(x^2 + 3y^2)) \\ &\quad + s_2(x^2 + y^2) \end{aligned} \tag{3}$$

(u_o, v_o) 는 영상 이미지의 중심 좌표를 나타내며 d_x 와 d_y 는 카메라 내부 변환 변수의 하나이며 $\delta^{(x)}(x, y)$, $\delta^{(y)}(x, y)$ 는 왜곡을 보정해주는 함수이고 k_1, p_1, p_2, s_1 및 s_2 는 왜곡 상수들이다. (1), (2) 및 (3)을 이용하여 (u, v) 와 (X, Y) 에 대한 식으로 나타내면 (4)로 나타낼 수 있다[14].

$$X = \frac{\sum_{0 \leq i+j \leq 3} a_{ij}^{(1)} u^i v^j}{\sum_{0 \leq i+j \leq 3} a_{ij}^{(3)} u^i v^j}, Y = \frac{\sum_{0 \leq i+j \leq 3} a_{ij}^{(2)} u^i v^j}{\sum_{0 \leq i+j \leq 3} a_{ij}^{(3)} u^i v^j} \quad (4)$$

여기서 $a_{ij}^{(k)}$ 는 변환 상수이다. 다시 (4)는 다음 형태로 바꿀 수 있다.

$$\rho x_w = A \hat{u} \quad (5)$$

$$A = \begin{pmatrix} a_{33}^{(1)}, \dots, a_{00}^{(1)} \\ a_{33}^{(2)}, \dots, a_{00}^{(2)} \\ a_{33}^{(3)}, \dots, a_{00}^{(3)} \end{pmatrix} \quad (6)$$

$$\hat{u} = (u^3, u^2v, uv^2, v^3, u^2, uv, v^2, u, v, 1)^T \quad (7)$$

여기에서 ρ 는 $a^{(3)} \hat{u}$ 이고 $a^{(3)}$ 은 A행렬의 세 번째 row 벡터이다. x_w 는 $(X, Y, 1)^T$ 로 전역 좌표 값을 나타낸다. 즉 이 행렬의 계수 값을 찾으면 왜곡을 고려한 좌표변환을 할 수 있다. 그림 11에는 3차원 데이터를 찾기 위한 폭이 2cm인 격자무늬를 나타내고 있다. 이 중 15개의 표본 데이터를 가지고 식 6에서의 행렬 A의 계수를 찾을 수 있다.

3. 신경회로망 응용

본 논문에서는 신경망을 이용하여 행렬 A의 계수 값들을 구하였다. 신경망으로 그림 11에서 찾은 15개 표본 데이터(표4)를 학습한 시킨 후에 학습이 끝나면 행렬 A 대신에 사용하였다.

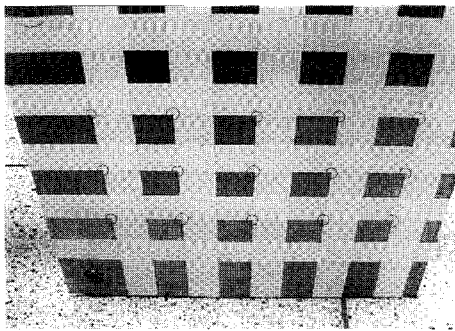


그림 11. 15개의 표본 영상 좌표 데이터.

Fig. 11. 15 Image exemplars.

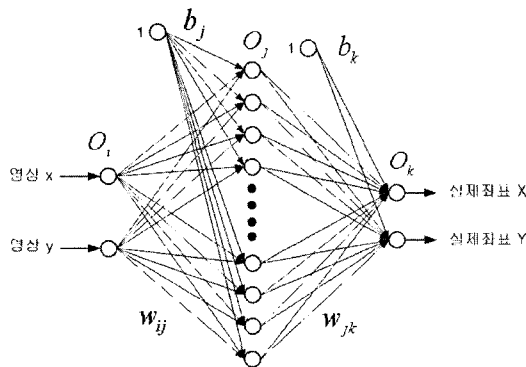


그림 12. 사용한 신경망 구조.

Fig. 12. Neural network structure.

그림 12는 사용한 신경망구조로 입력 유닛이 2개, 히든 유닛이 15개이고 출력 유닛이 2개인 다층 퍼셉트론을 이용하였다. 학습알고리즘은 역전파 알고리즘을 사용하였다. 비선형 함수로 hyperbolic tangent 함수를 사용하였기에 입력 영상의 좌표 값을 -5.0~5.0 사이에 오도록 입력하였다.

표 2는 (4)로 계산한 데이터이다. 행렬 A를 알고 있을 때 이미지 좌표 데이터를 넣은 결과이다. 상당히 정확한 값을 알 수 있다. 그리고 표 3은 신경망을 표 1의 데이터로 학습 후에 출력한 결과이다. 표 2의 데이터와 비슷하게 나타나는 것을 알 수 있다.

신경망의 학습 결과를 테스트하기 위해 그림 13에서 표시한 데이터를 가지고 테스트하였다. 그리고 이러한 데이터들은 그림 11에서 사용하지 않은 데이터이며, 영상이 중심에서 멀어질수록 왜곡을 많이 받기 때문에 그림 11에서의 데이터보다 왜곡이 심한 좌표들이다. 표 4는 그림 13에서 표시한 데이터를 나타내고 있다.

표 1. 표본 데이터.

Table 1. Exemplars.

실제값 X(m)	-0.0075	-0.035	0.005	0.045	0.085	-0.0075	-0.035	0.005
실제값 Y(m)	-0.14	-0.14	-0.14	-0.14	-0.14	-0.18	-0.18	-0.18
이미지 x	122	236	353	469	583	136	243	350
이미지 y	163	163	163	165	167	245	246	247
실제값 X(m)	0.045	0.085	-0.0075	-0.035	0.005	0.045	0.085	
실제값 Y(m)	-0.18	-0.18	-0.22	-0.22	-0.22	-0.22	-0.22	
이미지 x	458	563	150	249	348	448	545	
이미지 y	248	249	315	317	319	319	319	

표 2. (4)에 의한 표본 데이터 출력.

Table 2. Data from equation (4).

실제값 X(m)	-0.0075	-0.035	0.005	0.045	0.085	-0.0075	-0.035	0.005
실제값 Y(m)	-0.14	-0.14	-0.14	-0.14	-0.14	-0.18	-0.18	-0.18
계산값 X(m)	-0.0075	-0.035	0.005	0.045	0.085	-0.0075	-0.035	0.005002
계산값 Y(m)	-0.14	-0.14	-0.140000	-0.14	-0.14	-0.179469	-0.180005	-0.18007
실제값 X(m)	0.045	0.085	-0.0075	-0.035	0.005	0.045	0.085	
실제값 Y(m)	-0.18	-0.18	-0.22	-0.22	-0.22	-0.22	-0.22	
계산값 X(m)	0.045	0.085	-0.0075	-0.035	0.0050002	0.0450001	0.085	
계산값 Y(m)	-0.18	-0.180000	-0.219989	-0.22	-0.22	-0.22	-0.22	

표 3. 표본 데이터 학습 후 신경망 출력.

Table 3. Data from neural network.

실제값 X(m)	-0.0075	-0.035	0.005	0.045	0.085	-0.0075	-0.035	0.005
실제값 Y(m)	-0.14	-0.14	-0.14	-0.14	-0.14	-0.18	-0.18	-0.18
계산값 X(m)	-0.00750	-0.034999	0.0050	0.045007	0.084978	-0.007499	-0.0350	0.00500
계산값 Y(m)	-0.13995	-0.139974	-0.140034	-0.13998	-0.140004	-0.180102	-0.180026	-0.179932
실제값 X(m)	0.045	0.085	-0.0075	-0.035	0.005	0.045	0.085	
실제값 Y(m)	-0.18	-0.18	-0.22	-0.22	-0.22	-0.22	-0.22	
계산값 X(m)	0.04498	0.085047	-0.0075	-0.03500	0.004998	0.045007	0.084973	
계산값 Y(m)	-0.18002	-0.179998	-0.219945	-0.219998	-0.220032	-0.220009	-0.220003	

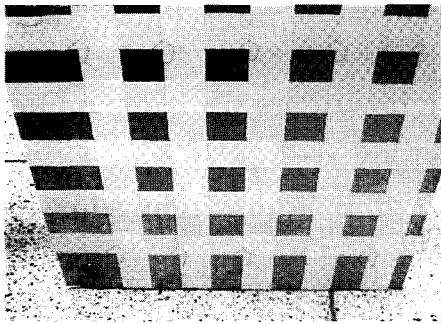


그림 13. 10개의 테스트 영상좌표 데이터.

Fig. 13. 10 image exemplars for test.

표 4. 테스트 데이터.

Table 4. Exemplars for test.

실제값 X(m)	-0.0075	-0.035	0.005	0.045	0.085
실제값 Y(m)	-0.1	-0.1	-0.1	-0.1	-0.1
실제값 X(m)	-0.0075	-0.035	0.005	0.04	0.085
실제값 Y(m)	-0.26	-0.26	-0.26	-0.26	-0.26

표 5. (4)에 대한 테스트 결과.

Table 5. Test results from equation (4).

점	실제값 X(m)	계산값 X(m)	오차 X(m)	실제값 Y(m)	계산값 Y(m)	오차 Y(m)
1	0.0075	0.002654	0.004846	0.10	0.043185	0.143185
2	0.035	0.008298	0.043298	-0.10	0.023081	0.076919
3	0.005	0.002205	0.002795	0.10	-0.066523	0.033477
4	0.045	0.034493	0.010507	0.10	-0.085388	0.014612
5	0.085	0.073799	0.011201	-0.10	-0.092439	0.007561
6	0.0075	0.002644	0.010144	-0.26	0.026279	0.233721
7	0.035	0.017028	0.017972	-0.26	0.144609	0.115391
8	0.005	0.002032	0.002968	-0.26	-0.123779	0.136221
9	0.045	0.412401	0.457401	-0.26	2.001109	2.261109
10	0.085	0.117924	0.032924	0.26	-0.346282	0.086282

표 6. 신경망에 대한 테스트 결과.

Table 6. Test results from neural network.

점	실제값 X(m)	계산값 X(m)	오차 X(m)	실제값 Y(m)	계산값 Y(m)	오차 Y(m)
1	0.0075	0.005663	0.001837	-0.10	0.111773	0.011773
2	0.035	0.033802	0.0012	-0.10	-0.112072	0.012072
3	0.005	0.006692	0.001692	0.10	0.11250	0.01250
4	0.045	0.046648	0.001648	0.10	0.111096	0.011096
5	0.085	0.084948	0.000052	0.10	0.110850	0.010850
6	0.0075	0.008088	0.000588	-0.26	0.250905	0.009905
7	-0.035	0.03590	0.0009	-0.26	0.249976	0.010024
8	0.005	0.003955	0.001045	-0.26	-0.249644	0.010356
9	0.045	0.044499	0.000501	0.26	-0.250084	0.009916
10	0.085	0.084552	0.000448	0.26	0.250742	0.009258

표 5와 표 6은 표 4의 데이터로 테스트한 각각의 결과이다. 표 2와 3에서는 어느 정도 매우 비슷한 데이터를 보여주지만 왜곡이 심한 부분에서는 오차가 생기는 것을 알 수 있다. 표 5는 (4)에 의한 데이터로 오차가 많이 나는 것을 알 수 있다. 이것은 영상의 왜곡을 충분히 반영하였다고 보기 어렵다. 표 6은 신경망으로 출력한 데이터로 오차가 최대 2mm를 넘지 않는다. X축의 경우에 측정범위가 약 10cm 정도로 2%의 오차이다. 그림 14와 15는 x와 y의 좌표에 대한 오차를 각각 나타낸다. 그림에서 왼쪽 바는 식에 의한 오차이고 오른쪽 바는 신경망을 통한 오차이다. 신경망의 경우 오차가 현저하게 작아지는 것을 볼 수 있다.

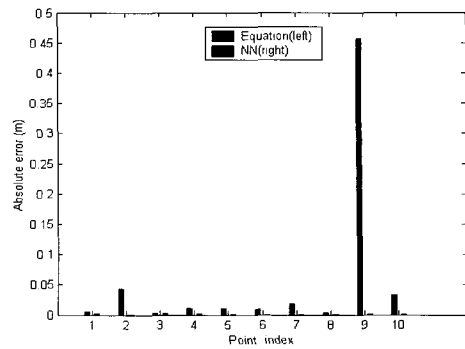


그림 14. x좌표의 오차 비교.

Fig. 14. Errors in x coordinate.

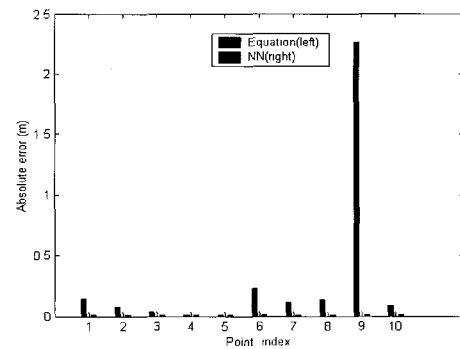


그림 15. y좌표의 오차 비교.

Fig. 15. Errors in y coordinate.

IV. 이동 로봇의 틈새 경로 추종

그림 16은 이동로봇과 틈새 경로에 대한 그림이다. 그림에서 알 수 있듯이 이동 로봇이 틈새를 추종하기 위해서는 어느 한 부분이 틈새를 계속 따라가면 된다. 즉 이동 로봇의 앞부분이나 작업 공간, 또는 바퀴축의 중심 등이 틈새를 추종하면 될 것이다. 하지만 본 논문에서 사용한 틈새 검출 센서부가 이동 로봇의 앞부분에 있기 때문에 앞부분이 틈새를 따라가도록 하였다. 그림 16에서 이동로봇의 제어는 무게 중심이 바퀴의 중심축의 연장선상에 있다고 가정하고 기구학적으로 제어를 하였다. 이동 로봇의 속도는 두 바퀴의 합과 비례하고 회전은 두 바퀴의 차와 비례한다. 이런 특성을 이용하면 (8)과 같이 이동로봇이 전체적인 속도는 일정하게 하면서 회전만 하는 PD 형태의 제어 방식을 생각할 수 있다.

$$\begin{aligned} \tau_r &= V_{desired} + (K_p \times \phi_e + K_d \times \dot{\phi}_e) \\ \tau_l &= V_{desired} - (K_p \times \phi_e + K_d \times \dot{\phi}_e) \end{aligned} \quad (8)$$

여기에서 $V_{desired}$ 는 원하는 이동 로봇의 직진 속도이며 ϕ_e 는 (9)에서 구한 값이다. 그리고 K_p 와 K_d 는 제어이득 값이다.

$$\phi_e = \tan^{-1}(S_e/l) \quad (9)$$

여기서 S_e 는 그림 16에 보여진 거리이며 l 은 바퀴축과 앞부분 사이의 거리를 나타낸다. 이와 같은 제어 법칙은 아래 그림 17에 블록 다이어그램으로 나타나 있다.

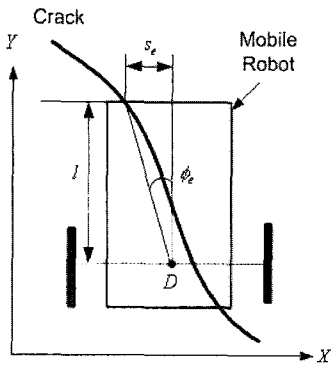


그림 16. 이동로봇과 틈새.

Fig. 16. Configuration of robot and crack.

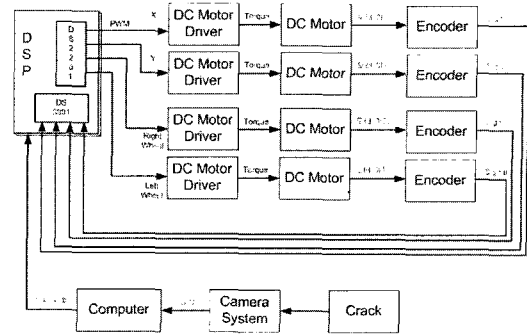


그림 18. 실험장치의 구성도.

Fig. 18. Overall structure.

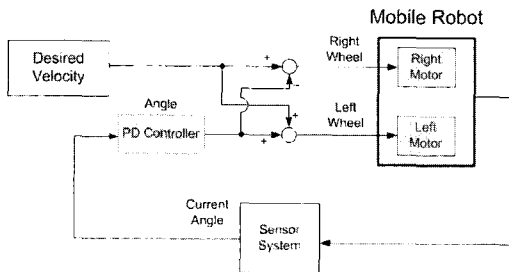


그림 17. 이동 로봇 제어 블록도.

Fig. 17. Control block diagram.

V. 실험 환경 및 결과

1. 실험 환경

대략적인 전체 시스템 구성은 그림 18에 실험장치 구성도를 나타내었으며, 실험은 TMS320C40이 내장된 dSPACE사의 DS1003 DSP 보드, DS2201 아날로그 입출력 보드, DS3001 엔코더 보드를 이용하여 제어기를 만들었고 샘플링은 1KHz로 수행하였다. 제안된 이론에 따라서 pwm 명령을 생성하여 모터 드라이버로 명령을 전달하면 모터 드라이버는 DC모터를 H-Bridge 회로를 이용하여 구동시켰다. XY 테이블은 모터에 붙어있는 풀리와 타이밍 벨트에 의해서 움직이고 이동로봇은 모터에 있는 바퀴에 의해서 이동한다. 테이블이 이동한 거리 측정은 모터 축에 연결되어 있는 엔코더를 사용하였고 이동로봇의 움직임도 바퀴 모터의 엔코더를 사용하였다. 틈새의 인식은 이동 로봇에 있는 컴퓨터에 의해 감지하여 DSP에 전달된다.

2. 틈새 추종 실험

그림 19는 이동 로봇이 틈새를 따라가는 실험 환경을 보여주고 있다. 그림에서 보여주는 것처럼 두 개의 합판을 이용하여 길이가 약 2.5m이고 폭이 약 1cm인 틈새를 임의로 만들어 실험을 하였다. 초기에 틈새가 이동 로봇의 오른쪽에 비스듬하게 위치시켰다. 로봇은 갈라진 틈새를 중심으로 추종하도록 되어 있다.

그림 20은 이동로봇의 위치 추종 실험 결과를 보여주고 있다. 처음에 이동로봇의 중심과 틈새와의 거리는 약 8cm 정도이다. 이동 로봇이 2초부터 시작해서 약 3초까지 빠르게 수렴해간다.

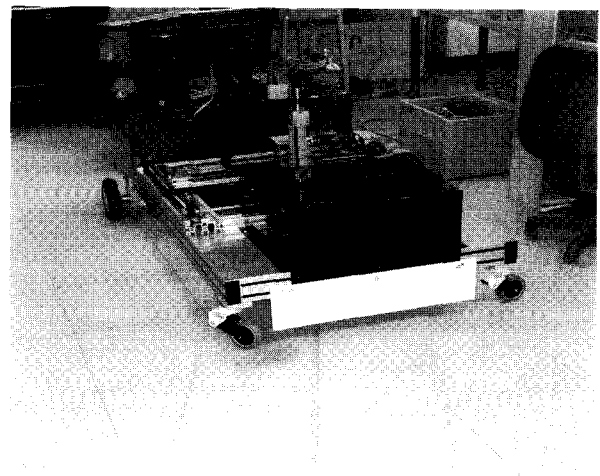


그림 19. 실험 환경.

Fig. 19. Experimental environment.

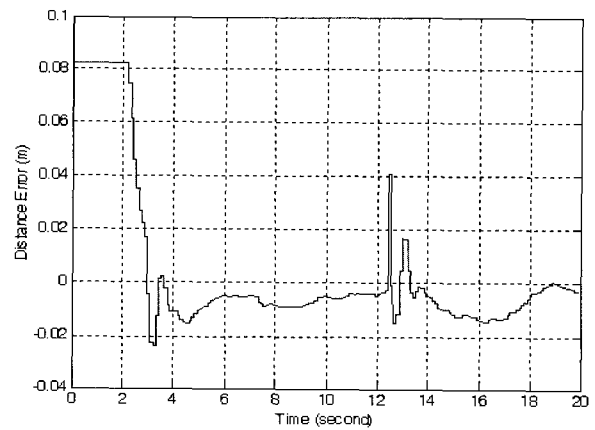


그림 20. 실험 결과.

Fig. 20. Tracking result.

그 후에 약간 진동을 하는데 이것은 이동로봇의 구조상 축의 구동에 의한 방향을 바꿈에서 오는 것으로 PD 제어기의 게인값에 의한 오버슈트 때문이다. 그리고 12초까지 약 -1cm 정도로 수렴함을 알 수 있다. 13초에 한 번 진동하는 것은 그림 19에서 오른쪽의 합관을 두 개로 겹쳐 놓아서 생긴 것으로 순간적으로 잘못된 틈새를 인식하여 생긴 결과이다. 하지만 틈새를 청소하는 작업을 할 경우의 허용오차 안으로 설정되어 오차의 허용 범위를 만족한다.

VI. 결론

본 논문에서는 틈새를 인식하고 추종해가는 이동 로봇을 제작하였다. 틈새를 인식하기 위한 시스템으로 레이저와 카메라 영상을 이용하였고, 2차원 영상 데이터의 문제점을 해결하기 위해 신경망의 학습을 이용한 3차원 데이터를 얻는 방법을 제시하여 좀 더 정확한 틈새 위치를 파악할 수 있었다. 틈새의 위치를 인식한 뒤에 제어 알고리즘을 적용하여 로봇이 틈새를 추종하도록 실험하였다. 틈새 추종 실험에서는 로봇이 잘 추종함을 볼 수 있었으나, 기구학 기반의 간단한 PD제어를 사용하므로 다소 추종 오차가 발생 하였다. 또한 내부환경에서 실험하였으나 야외의 실제 환경에서 실현하는 것이 필요하다. 현재는 틈새를 인식하여 로봇이 추종하는 단계에 있으나, 추후에는 XYZ 로봇이 틈새에 있는 이물질을 제거하고 틈새를 매우는 작업을 하도록 추가할 예정이다.

참고문헌

- [1] T. A. Lasky and B. Ravani, "Sensor-based path planning and motion control for a robotic system for roadway crack sealing", *IEEE Transactions on Control Systems Technology*, vol. 8, no. 4, pp. 609-622, July 2000.
- [2] Y. B. Jeon, B. O. Kam, S. S. Park, and S. B. Kim, "Seam tracking and welding speed control of mobile robot for lattice type Welding", *Proceedings. ISIE 2001. IEEE International Symposium on*, vol 2, pp. 857-862, June 2001.
- [3] R. S. Petty, M. Robinson and J. P. O. Evans, "3D measurement using rotating line-scan sensors", *Meas. Sci. Technol.* vol, 9, pp. 339-346, 1998.
- [4] Y.-H. Fa and Y. Moalem, "Effective defect detection and

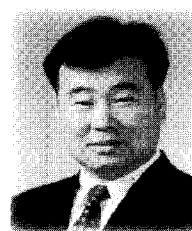
classification methodology based on integrated laser scanning inspection and automatic defect classification", *IEEE/SEMI Advanced Semiconductor Manufacturing Conference*, pp. 266-271, 1998.

- [5] X. Xu, C. S. Xu, Y. L. Tian, S. D. Ma and M. L.- Luo, "3-D face image acquisition and reconstruction system", *Technology Conference 1998*.
- [6] W. T. Woo and A. Ortega, "Modified overlapped block disparity compensation for stereo image coding", *SPIE* vol. 3635, pp. 570-581, 1999.
- [7] W. Czajewski and A. Sluzek, "Development of a laser based vision system for an underwater vehicle", *ISIE'99-Bled, Slovenia*, pp. 173-177, 1999.
- [8] D. S. Yang, J. H. Chen H. Zhou and S. Buckley, "Two practical ways to avoid spurious reflections from shiny surfaces on a 3D machine vision inspection system", *SPIE* vol. 3652, pp. 30-33, January 1999.
- [9] S. Tetlow and J. Spours, "Three-dimensional measurement of underwater work sites using structured laser light", *Meas. Sci. Technol.* vol. 10, pp. 1162-1167, 1999.
- [10] R. Fierro, and F. L. Lewis, "Control of a nonholonomic mobile robot using neural networks", *Proceedings of the IEEE International Symposium on Intelligent Control*, pp. 415-421, Aug. 1995.
- [11] T.-C. Lee, K.-T. Song, C.-H. Lee, and C.-C. Teng, "Tracking control of mobile robots using saturation feedback controller," *IEEE International Conference on Robotics and Automation*, pp. 2639-2644, May 1999.
- [12] W. Weiguo, C. Huitang, and W. Yuejuan, "Backstepping design for path tracking of mobile robots", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Volume 3, pp. 1822-1827, Oct. 1999.
- [13] J. Weng, P. Cohen, and M. Herniou, "Calibration of stereo cameras using a non-linear distortion model", *IEEE Conference on Pattern Recognition*, pp. 246-253.
- [14] J. Weng, P. Cohen, and M. Herniou, "Camera calibration with distortion models and accuracy evaluation", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 14, no. 10, pp. 965-980.



조 헌택

1974년 7월 7일생. 2000년 충남대학교 메카트로닉스공학과 졸업. 2003년 동 대학원 석사 졸업. 현재 기계연구원 지능 정밀 기계 부문 연구원. 관심 분야는 신경회로망 제어기 및 DSP 시스템.



정 슌

1964년 9월 11일생. 1988년 미국 웨인 주립대 전기 및 컴퓨터 공학과 졸업. 1991년 미국 캘리포니아대 데이비스 전기 및 컴퓨터 공학과 석사. 동 대학 박사. 1997년~현재 충남대학교 메카트로닉스공학과 부교수. 관심분야는 지능

제어 및 지능 로봇 시스템, 로봇과 인간, 무인 로봇의 위치 추정 및 제어.