

센서 합성을 이용한 자율이동로봇의 주행 알고리즘 설계

論 文
53D-10-5

Design of Navigation Algorithm for Mobile Robot using Sensor fusion

金正勳[†] · 金瑛中^{*} · 林 畝澤^{**}

(Jung-Hoon Kim · young-Joong Kim · Myo-Teag Lim)

Abstract - This paper presents the new obstacle avoidance method that is composed of vision and sonar sensors, also a navigation algorithm is proposed. Sonar sensors provide poor information because the angular resolution of each sonar sensor is not exact. So they are not suitable to detect relative direction of obstacles. In addition, it is not easy to detect the obstacle by vision sensors because of an image disturbance. In This paper, the new obstacle direction measurement method that is composed of sonar sensors for exact distance information and vision sensors for abundance information. The modified splitting/merging algorithm is proposed, and it is robust for an image disturbance than the edge detecting algorithm, and it is efficient for grouping of the obstacle. In order to verify our proposed algorithm, we compare the proposed algorithm with the edge detecting algorithm via experiments. The direction of obstacle and the relative distance are used for the inputs of the fuzzy controller. We design the angular velocity controllers for obstacle avoidance and for navigation to center in corridor, respectively. In order to verify stability and effectiveness of our proposed method, it is apply to a vision and sonar based mobile robot navigation system.

Key Words : Fuzzy, Mobile, Robot, Obstacle, Vision

1. 서 론

자율 이동 로봇의 동작을 위해서 가장 먼저 필요한 것이 위치 정보에 대한 인식과정이다. 작업 공간 안에서의 자기 위치 인식(self-localization)과 객체 인식(object recognition), 장애물 회피(obstacle avoidance)를 위해서는 주변 환경에 대한 정보를 다양한 센서를 이용해서 얻어내고, 이를 해석하여 주변 환경에 대한 정확한 지도를 만들고 이를 계속해서 갱신에 나가는 작업이 필요하다. 시각 센서를 통해 얻어진 영상 정보를 사용하여 정해진 로봇의 위치와 자세를 직접 제어하는 것을 시각 구동(visual servoing)이라고 한다 [1]. L. E. Sanderson과 A. C. Weiss는 [2]에서 시각 서보(visual servo) 구조에 있어서 크게 두 가지, 위치 기반 제어(position-based control)와 영상 기반 제어(image-based control)로 의미 있는 분류를 하였다 [1][3]. 영상 기반 제어는 특정 목표물의 정확한 위치 결정을 위한 3차원 영상 해석이 필요없고, 센서 모델링, 목표물과 로봇 모델링, 카메라 보정으로 인한 오차와 계산시간을 줄일 수 있는 장점으로 인해 연구의 많은 부분을 차지하고 있다. 또한 영상 기반

제어는 영상 추출 샘플링 주기 사이에 발생할 수 있는 위치 오차와 noise 에 안정하여 실시간 구현에 적합하다 [4]. 영상 기반 제어에서 영상 특징 자코비안은 L. E. Weiss 등에 의해 제안되었고 [2], 로봇의 움직임에 따른 영상 특징의 변화량 관계를 나타낸다. R. Kuc등은 박쥐가 음파를 이용하여 방향과 거리를 조정하면서 먹이를 잡는 동작 원리와 유사하게 이동 로봇이 초음파 센서를 사용해 위치인식을 시도하였다 [5]. 여기서는 로봇의 거리와 경로는 이미 알고 있다는 가정 하에서 이뤄졌다. P. M. Vaz등은 물건을 옮기는 동작을 위한 nonholonomic 이동 로봇과 저가의 적외선 센서들을 이용한 이동경로 생성 방법을 제안하였다 [6]. 이는 Encoder, laser, sonar 센서들을 통해 목표점 근처까지의 최단 경로를 선택하여 이동하면서 적외선 센서를 통해 물체를 인식하는 방법이다. 여기서 저가의 구현을 위해서 보다 더 정확하고 효율적인 시각(vision) 센서를 사용하지 않았음을 언급하고 있고 그 유용성에 대해 간단히 언급하고 있다. 따라서 보다 정확하고 견실한 주행을 위해 시각 센서가 필요함을 알 수 있다. R. C. Arkin [7]에서 영상 인식을 통한 주행 알고리즘을 제시하였다. 특히 보다 정밀하게 위치를 결정하기 위해 랜드마크의 모서리를 기반으로 삼차원 거리정보를 얻어 수행하였다. 여기서는 모노카메라를 사용하기 때문에 거리와 방향을 결정하기 위한 알고리즘이 필요하여 많은 계산량과 정확성 확보를 위한 오차 보정이 요구된다. 따라서 알고리즘 개선을 위해서 거리와 방향 정보를 위한 스테레오 카메라의 사용이 유용함을 알 수 있다. 그러나 캘리브레이션이나 모델링에 따른 오차 발생 문제는 남아 있다. N. M. Charkari등 [8]은 주행을 위한 전략으로서 CCD 카메라

[†] 교신저자, 正會員 : LG전자 연구원

E-mail : kjoin@hanmail.net

* 正會員 : 고려대학교 전기공학과 박사과정

** 正會員 : 고려대학교 전기전자전파공학부, 부교수

接受日字 : 2004年 7月 5日

最終完了 : 2004年 8月 23日

라로 바닥에 있는 검은 줄무늬를 찾고 초음파 센서로 벽면의 거리를 측정하여 주행하는 연구를 제안했다. 여기서는 자기의 위치를 확인하기 위해서만 시각(vision) 센서를 사용하고 주행할 때에는 초음파 센서를 사용하는데 이는 인위적으로 주어진 한정된 공간에서 사용할 수밖에 없고 센서의 불확실성에 따라 제어하는데 다소 어렵다. I. Mitsuru 등[9]은 optical flow를 사용하여 로봇의 동작을 영상으로부터 직접 제어하였다. 여기서는 optical flow 파라미터를 통하여 제어하게 되는데 계산 시간이 다소 요구되어 영상 처리 시간 사이에 오차가 벌어지는 문제점(aperture problem)이 생긴다. 이런 문제점은 실시간 구현에 있어서 부적절한 결과를 가져오게 되므로 이에 적합한 방법이 필요하다.

본 논문에서는 CCD 카메라에서 얻어진 2차원 영상 정보와 sonar 센서에서 얻어진 거리 정보를 효과적으로 합성하기 위한 방법에 대해서 연구하였다. 이때, 카메라에서 얻어진 이미지 정보를 전처리 과정에 있어서 주변 환경의 조명에 따라서 다른 결과를 얻게 될 수도 있고, 주변 환경과 구분해 내고자 하는 대상의 색상과 밝기가 유사한 경우에는 임계값을 결정하는데 어려움을 겪을 수도 있으며, 에지 분포의 특성을 이용하기 위해서는 주변 환경이 규칙적인 형태를 가져야만 하는 등의 제약 조건을 가지게 된다. 이를 해결하기 위해서 별도의 전처리 과정과 주변 환경에 대한 제약 없이 적용시킬 수 있는 영상 통합/분리(image merge/split) 모델을 제안한다. sonar 센서에서 얻어지는 거리 정보의 이미지 평면에서의 대응관계를 살펴보기 위해서, 이미지 평면의 각 픽셀들이 대표하는 각도 성분을 계산하여 이미지 픽셀과 각도 성분 사이의 직접적인 대응 관계를 설정한다.

영상을 통해서 얻어진 장애물 정보와 sonar 센서 모듈에서 얻어진 로봇의 주행위치 정보를 토대로 로봇의 주행속도 및 방향정보를 계산하는 퍼지 제어기에 대해서 연구하였다. 본 논문에서는 영상을 통해서 얻어진 장애물의 위치정보와 sonar 센서 모듈에서 얻어진 로봇의 주변 거리 정보를 토대로 장애물회피와 안전한 주행위치를 찾는 개선된 퍼지 주행제어기를 제안하고, 시뮬레이션 작업을 통하여 제안된 제어기의 성능을 입증한 후, 제안된 제어기를 사용하여 실제 건물의 복도에서 실험함으로써 타당성을 보였다.

2. 본론

로봇이 주행하는데 필요한 정보를 얻어서 처리하기까지의 전체적인 흐름도는 그림 1과 같다. 그림에서 짧은 점선 지역은 장애물 인식 시스템을 나타내고, 긴 점선 지역은 로봇 주행 제어기를 나타낸다. 장애물 인식은 영상처리를 통해서 이루어지며, 카메라를 통해 들어온 영상과 sonar 센서에서 얻어진 거리 정보를 융합하여 장애물의 유·무 및 위치를 판단한다. 로봇 주행 제어기는 영상처리에서 검출된 장애물의 위치 정보와 sonar 센서에서 얻은 복도의 거리차와 장애물의 거리 정보를 가지고 로봇의 주행 속도 및 각속도를 계산하는 기능을 한다.

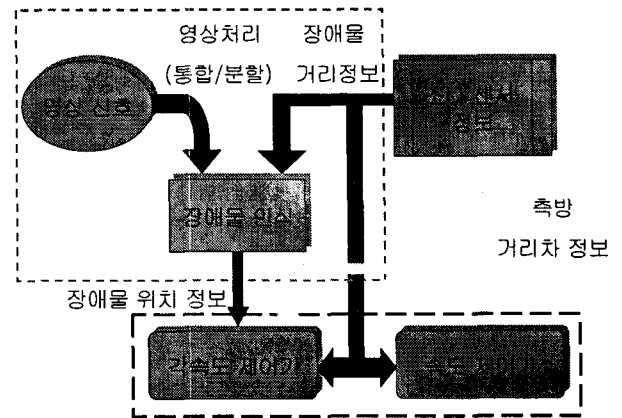


그림 1 로봇의 신호 흐름도
Fig. 1 Robot's signal flow chart

2.1 장애물 인식 방법

카메라로부터 들어온 영상신호와 sonar 센서로부터 획득한 정보를 조합하여, 장애물의 위치를 인식하는 방법에 대하여 알아본다. 그림 2는 장애물 위치 인식을 위한 순서도이다.

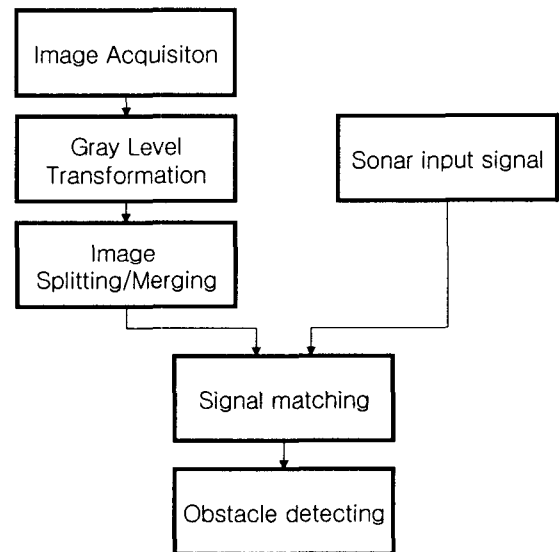


그림 2 장애물 위치 인식 순서도
Fig. 2 Flowchart of obstacle recognition

영상신호는 카메라로부터 들어온 신호를 말하며, PC에 장착된 프레임 그래버 카드를 이용해서 디지털 신호로 변환된다. 이 신호는 몇 가지의 영상처리 방법을 통하여 장애물의 위치 판별을 위한 신호로 사용된다. 초음파 신호는 발사된 신호의 TOF(time of flight)을 계산하여 거리를 측정한다. 초음파 센서에서 얻은 거리를 영상의 y축 픽셀값으로 변환하여, 동일한 y축 상에 존재하는 영상의 독립된 영역을 분리한다. 이는 실제 장애물의 거리와 방향을 획득하는데 사용된다.

2.1.1 영상 정보 처리

영상처리는 장애물의 거리 및 방향 정보를 획득하기 위해 필요한 작업이다. 일반적으로 물체를 인식하는데, 다양한 마스크를 이용한 에지 검출 방법이 많이 사용된다. 그러나, 이 방법은 물체외곽이 항상 폐곡선이 이루어지지 않으며, 이를 위해서 중심점을 분석한 후, 폐곡선을 형성하여 내부를 단일화하는 복잡한 과정을 거치게 된다. 따라서, 구현이 복잡하고, 계산시간이 오래 걸려, 실시간 처리에는 적합하지 않다. 본 논문에서 제안하는 분할/통합 알고리즘은 입력되는 영상 데이터 중 관심 부분인 장애물 정보만을 빠르고 효율적으로 검출하기 위해 전체영상의 동질성을 조사하여 영상을 분할/통합하는 방법이다. 기본적인 처리방법은 다음과 같다.

알고리즘 1. 물체인식을 위한 분할/통합 영상처리 방법

Step 1. 식 (4.1)을 이용하여 영역 R_i 에 대한 동질성 검사를 한다.

(a) 영역 R_i 이 동질성 ($P(R_i)=FALSE$) 이면 R_i 을 4개의 영역으로 나눈다. 분할된 영역에 대한 동질성 검사를 반복한다.

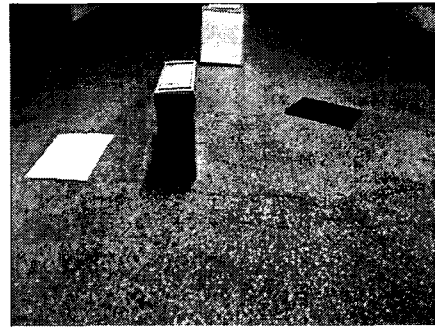
(b) 영역 R_i 이 동질성 ($P(R_i)=TRUE$) 을 가지면 영역전체의 intensity를 그 영역의 평균값으로 설정한다.

Step 2. 1단계의 분할과정이 끝나면 N개의 intensity로 경계값을 정하여 이웃한 영역들 사이에 동질성을 갖는 ($P(R_i \cup R_j)=TRUE$) 특성을 갖는 두 영역 R_i, R_j 을 통합한다.

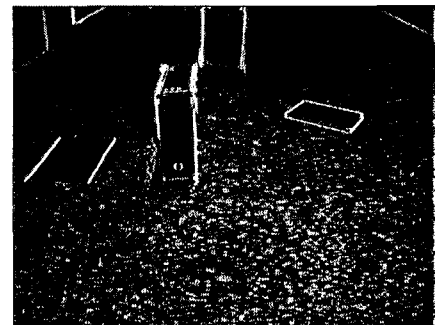
1단계 처리 과정이 끝나고 이웃 영역과의 분할/통합을 반복하여 전체 영상을 정리하는 기존의 방법은[10] 구현이 복잡하고 처리시간이 오래 걸려 자율이동로봇에 사용하기에는 부적합한 면이 있으므로, 본 논문에서는 변형된 분할 알고리즘을 사용한다. 1차로 분할된 영상을 N개의 경계값을 기준으로 영상을 재분할하면 효율적인 통합 알고리즘을 갖게 되고, 그 식은 다음과 같다.

$$P(R_i) = \begin{cases} TRUE & \text{if } f_{\max}(x, y) - f_{\min}(x, y) < T \\ FALSE & \text{if } f_{\max}(x, y) - f_{\min}(x, y) > T \end{cases} \quad (1)$$

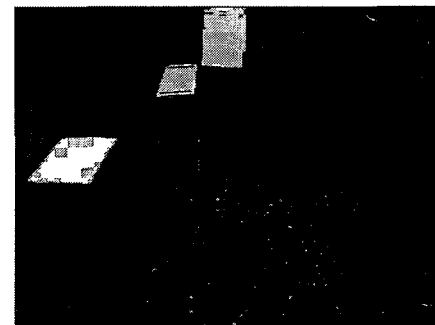
여기서, T 는 경계기준값을 나타내고, f_{\max}, f_{\min} 는 영역에서 intensity가 가장 작은값, 큰값을 나타낸다.



(a) 원영상



(b) 윤곽선 검출 영상



(c) 분할/통합 알고리즘 영상

그림 3 영상 처리 화면

Fig. 3 Image processing pictures

그림 3는 장애물이 있는 영상과 이를 소벨 마스크와 분할/통합 알고리즘에 적용한 영상이다. (b) 영상을 살펴보면, 장애물이 폐곡선을 이루지 않아, 단일영역으로 처리하기 곤란하고, 복도영역에 많은 왜란이 발생해 있다. 반면 제안한 분할/통합 알고리즘을 적용한 영상 (c)를 보면 장애물의 영역이 한가지 Gray Level로 처리되어 영상처리하기 적합하고, 왜란 또한 (b)영상에 비하여 현저히 적다. 이러한 장점을 토대로 제안한 알고리즘이 장애물 인식에 적합한 처리방법임을 알 수 있다.

2.1.2 영상과 실제 공간과의 관계

영상과 실제공간과의 관계는 그림 4와 같다.

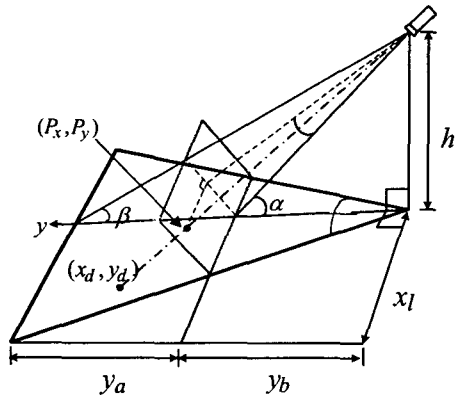


그림 4 영상과 실제공간과의 관계
Fig. 4 Relation of image and real world

카메라의 각도를 지면과 수평하게 놓지 않고, 경사를 이룬 것은 카메라가 획득한 영상과 sonar의 거리 정보를 매칭시키기 위함이다. 영상의 각 픽셀들은 실제공간으로 변환될 수 있는데, 이는 식 (2), (3) 로 표현된다.

$$y = h \tan \left[(90^\circ - \alpha) + \left(\frac{P_y}{S_y} \right) (\alpha - \beta) \right] \quad (2)$$

$$x = y \left(\frac{P_x}{S_x} \right) \tan \beta \quad (3)$$

여기서, S_x , S_y 는 영상좌표의 전체 픽셀수를 나타낸다.

2.1.3 조합한 장애물 인식 방법

sonar 센서의 측정값은 장애물과의 거리를 나타내므로, y 축 좌표로 표현될 수 있으며, 이는 식(2)를 변환한 식(4)와 같다. (2)식에서 얻은 y 축 좌표값은 영상에서 장애물을 찾는데 사용되고, 식(4)를 통해서 영상에서 장애물을 나타내는 영역을 찾을 수 있다.

$$P_{y_{sonar}} = \frac{S_y \left[\tan^{-1} \left(\frac{y_{sonar}}{h} \right) - (90^\circ - \alpha) \right]}{\alpha - \beta} \quad (4)$$

$$R_{obstacle}(P_x, P_y) = R_f(P_x, P_y) \ni P_{y_{sonar}} \quad (5)$$

$$(P_x \leq P_x \leq P_x, R_f(x, y) \neq R_{corridor})$$

여기서, P_x, P_x 는 영상에서 sonar S_i 의 측정영역 중 x 축의 좌·우측 가장 자리값을 나타내고, $R_{corridor}$ 는 복도영역을 나타낸다. 이렇게 얻어진 장애물영역은 식(6)를 이용하여 영상에서의 중심점을 구하고, 구해진 좌표값을 식 (2), (3)에 대입함으로써 측정위치로부터 장애물의 거리와 방향을 계산할 수 있다.

알고리즘 2. sonar 정보와 영상정보의 조합 방법

- Step 1. 제안된 분할/통합 처리를 한 영상을 Labelling 한다.
- Step 2. sonar 센서가 측정한 거리와 대응되는 y 축 픽셀의 Label Number를 알아본다. Label Number를 갖는 픽셀집합이 장애물 영역이다.
- Step 3. 식(6)을 계산하여 인식된 장애물의 x 축의 중앙값을 구하고, (5)를 이용하여 가장자리 값을 계산한다.

$$P_{x_{center}} = \frac{\sum_{k=1}^n P_{x_k}}{n}, P_{y_{center}} = P_{y_{sonar}} \quad (6)$$

$$(R_{obstacle} \ni P_{x_i}, P_{y_{sonar}})$$

여기서, n 은 장애물 영역의 픽셀수를 나타낸다. 이러한 장애물 인식 방법은 기존의 방법에 비해, 정해진 모양이 아닌 불특정한 모양의 장애물 인식이 가능하고, 장애물의 3차원 처리를 위해서 카메라를 2대 사용할 필요가 없다. 또한, 알고리즘이 단순하여, 구현이 쉽고 실시간 처리에 적합하다.

2.2 로봇 주행 제어기 설계

로봇을 주행시키는데 있어서 본 논문에서는 속도와 각속도 정보를 취한다. 속도는 장애물의 거리와 로봇의 측방 거리차를 이용하여 구하고, 각속도는 2 단계 작업을 거치는데, 1단계는 장애물의 거리와 영상처리해서 얻은 장애물의 위치를 가지고 장애물 회피 각속도를 구하고, 장애물의 거리와 로봇의 측방 거리차를 이용하여 로봇의 복도 중앙 주행에 필요한 각속도를 구한다. 2 단계로 두 개의 각속도 정보를 퍼지 제어 규칙을 만들어서 하나의 각속도 정보로 통합한다.

2.2.1 속도 제어기 설계

로봇의 속도는 전방의 장애물이 멀수록, 복도의 정 중앙을 주행할 때 증가한다. 로봇은 전방에 장애물이 나타나면 충돌회피를 위해 방향을 전환해야하고, 복도의 중앙을 주행하는 것이 안전하기 때문이다.

$$d(k) = \text{measure distance}(\text{sonar.No } 3,4) \quad (7)$$

$$dd(k) = \text{distance difference}(\text{sonarpair } 1-6, 2-5) \quad (8)$$

여기서, $d(k)$ 는 3,4번 소나의 측정 거리 중에서 최소 거리를 나타내고, $dd(k)$ 는 측면 소나쌍의 측정값들 중에서 그 차이가 가장 큰 값을 나타낸다. 이를 토대로 퍼지 제어기의 제어 규칙에 대한 Lookup table을 작성하면 표 1 과 같다.

표 1 속도 제어기의 제어 규칙

Table. 1 Lookup table of velocity controller

$dd(k)$ \ $d(k)$	N	Z	P
Near	S	M	S
Far	M	F	M

여기서 퍼지 제어기의 입력 변수는 전방향 sonar(3, 4번 sonar)의 측정거리(d), 측방향 sonar쌍(1-6, 2-5번 sonar) 중에서 가장 작은 값을 갖는 sonar의 sonar쌍의 값(dd)이고(예를 들면 6번 sonar의 값이 가장 작다면 1-6번 sonar 쌍의 차를 말한다.), 출력 변수는 로봇의 속도값(v)이다. $d(k)$, $dd(k)$, $u(k)$ 이 갖는 Fuzzy Label의 의미는 다음 표 2와 같다.

표 2 속도 제어기의 언어 변수

Table. 2 Linguistic variables of velocity controller

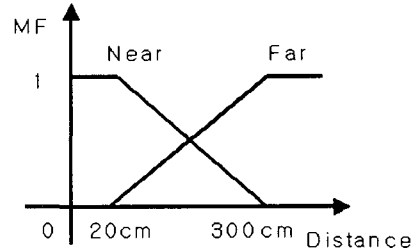
Fuzzy Label	Meaning
Near	Near
Far	Far
N	Negative
Z	Zero
P	Positive
S	Slow
M	Middle
F	Fast

그림 5는 속도 제어기에 사용된 입력과 출력에 대한 각각의 소속 함수(membership function)를 나타낸 것이고, 이들은 삼각형 형태를 통해 퍼지화 하였다.

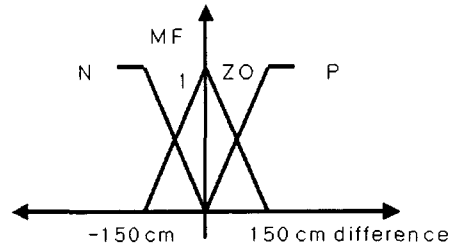
이렇게 설계된 속도 제어기는 기존의 장애물과의 거리만을 사용해서 속도를 계산하거나, 처음부터 속도를 상수로 설정해 놓고 주행했을 때보다 장애물과의 충돌 회피 능력을 좋게 하고, 로봇의 방향 전환시 바퀴의 미끄러짐을 방지한다.

2.1.1 각속도 제어기 설계

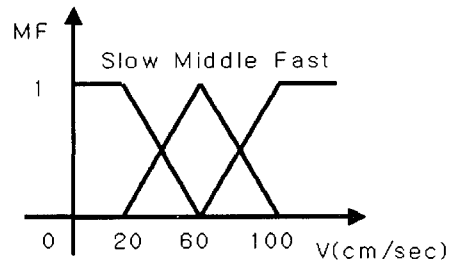
속도 제어기에서는 로봇의 전방 장애물과 측방 거리차를 이용했지만 각속도 제어기는 전방 장애물로 인한 회피 각속도와 측방 거리차에 의한 회피 각속도를 각각 구하고, 서로의 상관관계를 통해서 퍼지 제어기를 구현한다.



(a) 측정거리 소속 함수



(b) sonar쌍의 차 소속 함수



(c) 로봇의 속도 소속 함수

그림 5 속도 제어기의 소속 함수

Fig. 5 Membership functions of velocity controller

(1) 장애물 회피 각속도

전방 장애물로 인한 회피 각속도는 장애물의 거리와 영상 센서에서 얻은 장애물의 위치 정보를 이용해서 구한다.

$$d_{Obstacle}(k) = measure\ distance(sonar.No3,4) \quad (9)$$

$$a_{Obstacle}(k) = deg\ ree(R_{obstacle}) \quad (10)$$

여기서 $d_{Obstacle}$ 은 영상 처리에 의해 찾아낸 장애물을 sonar의 측정거리로 나타낸 것이고, $a_{Obstacle}$ 은 로봇의 중심 축과 가장 가까운 장애물의 측면이 이루는 각이다. 두 변수는 장애물을 회피하기 위한 각속도 퍼지 제어기의 입력 변수가 된다. 이를 토대로 퍼지 제어기의 제어 규칙을 작성하면 표 3과 같다.

표 3 장애물 회피 제어 규칙

Table. 3 Lookup table of obstacle avoidance

$a_{Obstacle}$ \ $d_{Obstacle}$	N	Z	P
Near	PB	ZO	NB
Far	PS	ZO	NS

$av_{Obstacle}$ 은 퍼지 제어기의 출력 변수로, 장애물을 회피하기 위한 각속도이다. 위의 입·출력 변수에 사용되는 Fuzzy Label은 표 4와 같다.

표 4 각속도 제어기의 언어 변수

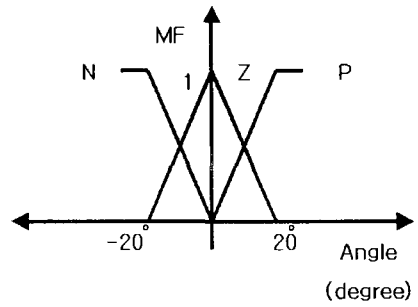
Table. 4 Linguistic variables of angle velocity controller

Fuzzy Label	Meaning
NR	Near
FR	Far
N	Negative Angle
Z	Zero Angle
P	Positive Angle
NB	Negative Big
NS	Negative Small
ZO	Zero
PS	Positive Small
PB	Positive Big

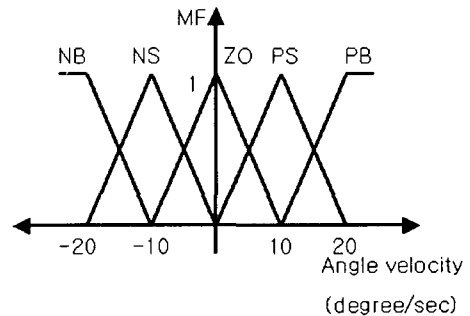
그림 6은 각속도 제어기에 사용된 입력 $a_{Obstacle}$ 과 출력 $av_{Obstacle}$ 에 대한 각각의 소속 함수(membership function)를 나타낸 것이고, 이들은 삼각형 형태를 통해 퍼지화 하였다.

(2) 측방 거리차 각속도

로봇의 방향 전환에 영향을 미치는 또 다른 요소는 로봇의 복도 중앙 주행이다. 로봇이 복도의 한쪽 측면으로 치우쳐서 주행하면 장애물이 나타나거나 복도벽면의 변화 발생시에 충돌의 위험에 효과적으로 반응하지 못한다. 복도 중앙 주행을 위한 각속도 제어기는 sonar 센서가 측정한 측방 거리차와 측정 거리중 가장 낮은 측정값을 사용해서 설계한다.



(a) $a_{Obstacle}$ 의 소속 함수



(b) $av_{Obstacle}$ 의 소속 함수

그림 6 각속도 제어기의 소속 함수

Fig. 6 Membership functions of angle velocity controller

$$d_{corridor}(k) = \min(\text{sonar No. } 1, 2, 5, 6) \quad (13)$$

$$dd_{corridor}(k) = \text{measure difference}(\text{sonar pair } 1-6, 2-5) \quad (14)$$

여기서 $d_{Corridor}$ 는 1,2,5,6번 sonar값 중에서 가장 작은 값을 가지는 측정값이고, $dd_{Corridor}$ 는 두 개의 sonar쌍 중에서 가장 작은 값이 속한 sonar쌍의 측정차를 말한다. 두 변수는 복도 중앙 주행을 위한 각속도 퍼지 제어기의 입력 변수가 된다. 이를 기준으로 퍼지 제어기의 제어 규칙을 작성하면 다음과 같다.

표 5 복도 주행 제어 규칙

Table 5 Lookup table of corridor navigation

$dd_{Corridor}$ \ $d_{Corridor}$	N	Z	P
Near	PB	ZO	NB
Far	PS	ZO	NS

$av_{Corridor}$ 는 퍼지 제어기의 출력 변수로 복도 중앙 주행을 위한 각속도이다.

이 제어기는 장애물 회피 각속도 제어기와 구성이 같고 입력 변수에서 장애물의 방향대신 복도의 거리차를 사용한다는 것만 다르다. 즉, $d_{Corridor}$ 와 $dd_{Corridor}$ 의 소속 함수는

각각 그림 3 (a), (b)와 같고. av Corridor 의 소속 함수는 그림 4 (b)와 같다.

(3) 통합된 각속도 제어기의 설계

위에서 설계된 두 개의 각속도 제어기는 통합되어 하나의 제어기로 설계되는데, 두 개의 출력 변수를 입력 변수로 사용하여 퍼지 제어기를 설계한다. 퍼지 제어기의 제어 규칙은 다음과 같이 나타낸다.

여기서, 출력 변수 av 는 최종적인 로봇의 각속도이다. 입력력과 출력에 사용된 변수로 이루어진 소속 함수의 형태는 그림 4 (b)와 같다.

표 6 최종적인 각속도 제어 규칙
Table 6 Lookup table of final angle velocity

av Obstacle \ av Corridor	NB	NS	ZO	PS	PB
NB	NB	NB	NB	ZO	ZO
NS	NB	NS	NS	ZO	ZO
ZO	NB	NS	ZO	PS	PB
PS	ZO	ZO	PS	PS	PB
PB	ZO	ZO	PB	PB	PB

통합된 제어기는 기존의 각도를 변환하는 제어기와 달리 주행 중에 방향의 전환이 가능하고 장애물 회피와 복도의 중앙 주행에 따른 상황을 분리해서 설계하였기 때문에 장애물에서 얻은 입력 변수와 sonar에서 얻은 입력 변수를 한꺼번에 이용해서 하나의 제어기를 설계했을 때 보다 제어 알고리즘이 단순하고, 계산량이 적다. 장애물이 없을 때 고속 주행과 장애물을 발견했을 때 효율적인 회피 경로를 얻을 수 있다. 제시된 제어기의 결과는 컴퓨터 시뮬레이션과 실제 주행 실험을 통해서 효용성을 입증한다.

2.3 실험 및 시뮬레이션

2.3.1 실험 장비

그림 7은 실험에 사용된 ACTIVMEDIA사의 Pioneer2-AT 로봇으로, 4륜 구동형 로봇이며, 몸체 주위에 16개의 초음파 센서를 장착하고 있다. 또한 이동로봇의 제어 및 알고리즘의 수행을 위하여 PC가 장착되어 있으며, 비전 시스템을 구성하기 위해 CCD camera 1대와 프레임 그래버로 Digital Translation사의 DT-3132를 장착하였다. 프로그램은 window98 기반의 visual C++을 사용하였다.

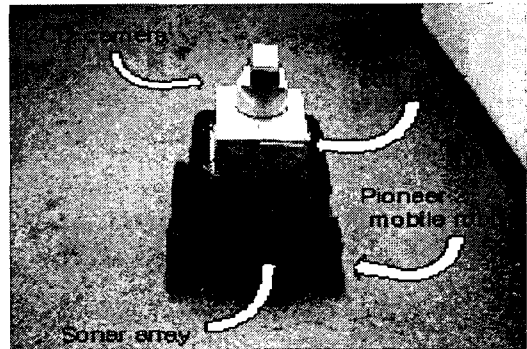
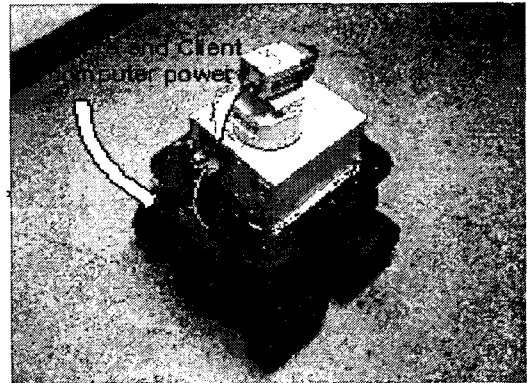


그림 7 실험에 사용된 Pioneer2-AT 로봇
Fig. 7 The experimental mobile robot

2.3.2 장애물 인식 실험

장애물 측정을 위해 위에 언급된 식에 사용될 상수값은 표 7과 같다.

표 7 측정 상수값
Table 7 Measurement parameter

h	S_y	S_x	α	β
62cm	480	640	45.94°	8.6°

실험을 위하여, ACTIVMEDIA사의 Pioneer2-AT를 사용했으며, sonar 센서는 전방의 3번 센서를 사용하였다. 그림 2의 영상에서 장애물의 sonar 측정 거리는 119.5cm로 측정되었으며, 대응되는 영상에서의 y 축 좌표값은 식(2)에 의해 구해지며, 그 값은 238point 이다. 이 값은 위에서부터 242point 이며, 장애물의 y 축 영역에 속해 있으며, 결국 $P_{y_{sonar}}$ 로 사용된다. 또한, sonar가 측정하는 영역은 x 축 좌표로 175~280 point로 측정되었다. 최종적으로 영상에서 식(4)를 만족하는 영역은 그림 4와 같으며, 중심점 좌표는 식(6)을 이용하면, x 축 좌표는 245 point 가 된다.

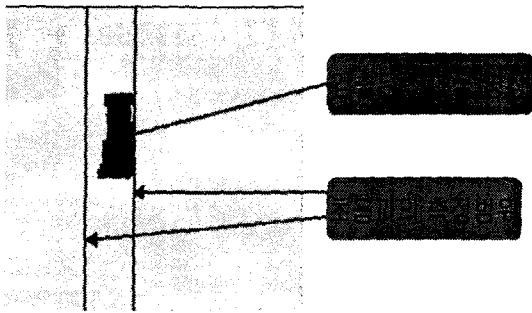


그림 8 장애물 검출 영상
Fig. 8 Obstacle's detecting image

그림 8에서, 두 개의 선 안쪽 영역이 사용된 sonar의 측정 영역이며, sonar는 이 영역 안에서 물체를 분간할 수 없고, 경계선에 걸쳐 있는 물체와 안에 있는 물체를 분간할 수 없으나, 이와 같은 영상처리 후에는 물체의 중심점좌표를 구하여 물체의 방향도 알 수 있다.

2.3.3 주행 제어기의 시물레이션

본 절에서는 2.2에서 설계한 속도 및 가속도 제어기에 대한 시물레이션을 하고, 시물레이션 결과에 대한 평가를 한다.

(1) 속도 제어기

그림 9는 Matlab을 이용해서 제어기를 구현한 후에 최종 결과 값인 로봇의 속도를 시물레이션 한 그림이다.

그림에 나타나는 표현에서 robot-velocity는 제어기의 출력 변수인 로봇의 속도이고, corridor-difference는 로봇을 기준으로 양 측면의 거리차를, obstacle-distance는 로봇의 전방에 나타나는 장애물과 로봇의 거리를 말한다.

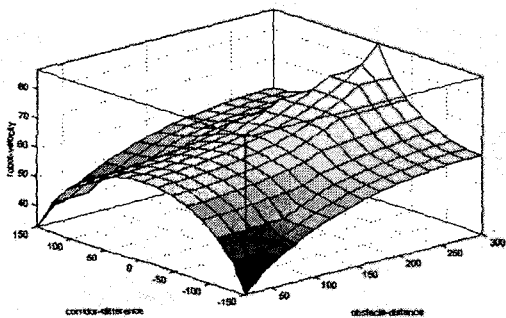


그림 9 속도 퍼지 제어기 결과
Fig. 9 Simulation of velocity fuzzy controller

로봇의 주행 중에 양 측면의 거리 차에 너무 민감하다 보면 로봇이 구불구불거리면서 주행하는 현상이 발생하고, 속도의 민감성을 너무 줄이면 장애물과의 충돌 위험성이 증가한다. 그림 9를 자세히 살펴보면, 저속이나 중속 주행 중에 양 측면의 거리차가 비교적 적으면, 속도의 변화량이 적어 주行的 효율성을 증가시켰고, 고속 주행 중에 측면 거리의

변화량이 발생하면 속도의 변화량을 증가시켜, 충돌의 가능성을 줄였다.

(2) 각속도 제어기

그림 10은 Matlab을 이용해서 제어기를 구현한 후에 최종 결과 값인 로봇의 각속도를 시물레이션 한 그림이다.

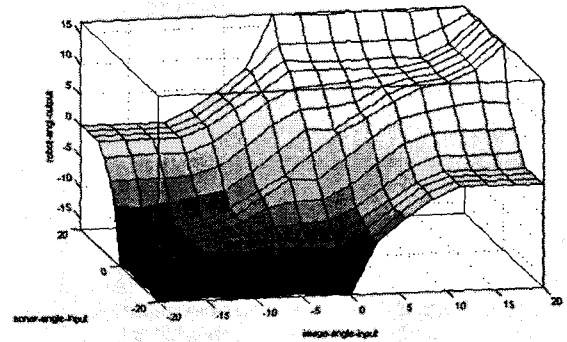


그림 10 각속도 퍼지 제어기의 결과
Fig. 10 Simulation of angle velocity fuzzy controller

그림에 나타나는 표현에서 robot-velocity는 제어기의 출력 변수인 로봇의 속도이고, corridor-difference는 로봇을 기준으로 양 측면의 거리차를, obstacle-distance는 로봇의 전방에 나타나는 장애물과 로봇의 거리를 말한다.

복도 주행 중에 장애물을 만나면 로봇은 복도의 중앙 주행과 장애물의 회피를 위한 동작을 취해야하는데, 이 두 동작이 서로 상반된 각속도의 변화를 요구하는 경우가 생긴다. 위의 시물레이션 결과를 살펴보면 장애물 회피와 복도 중앙 주행을 위한 제어기에서 각각의 출력을 얻어내어 퍼지 입력으로 사용함으로써 로봇의 각속도가 변하도록 하였다. 이러한 처리방식은 기존의 sonar 센서에 의지한 장애물 인식 방법이 범할 수 있는 극부 최소 상태(Local minimum)에 빠질 위험성을 줄여 준다.

2.3.4 주행 실험

본 절에서는 설계된 주행제어기를 실제 주행을 통해 그 성능을 입증한다. 이를 위하여 2가지 시나리오를 실험하였다. 실험 장소는 연구실 복도이며, 실험 장비는 장애물 인식 실험에 사용된 로봇과 영상 장비를 사용하였다. 속도 및 각속도는 평균적으로 400ms 간격으로 산출하여 로봇에 적용하였다. 비교 제어기로는 소나 센서만을 사용한 [11] 퍼지 제어기를 사용하였다. 이 제어기는 고정된 속도로 로봇을 주행시키며 그 속도는 40cm/sec이다. 또한 방향 전환방식은 각속도 변화가 아닌 제어기에서 계산된 각도를 변화시켜 방향을 전환시키며, 이때 angle velocity는 20cm/sec이다.

(1) 전방 장애물이 없을 때의 주행

<그림 11> 은 실험이 행해진 복도의 상황을 그림으로 나타낸 것이다.

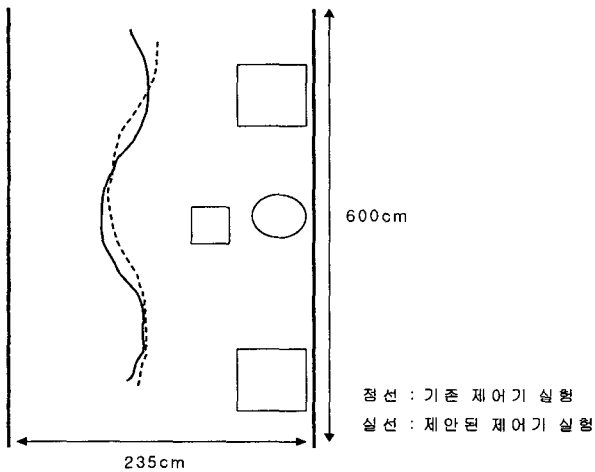
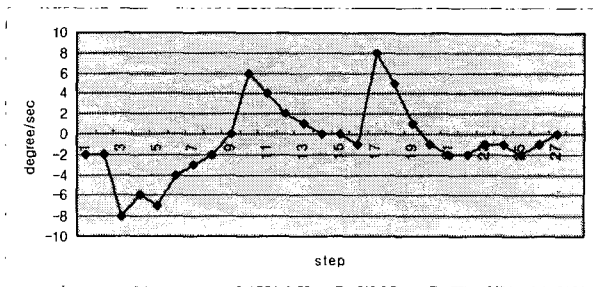
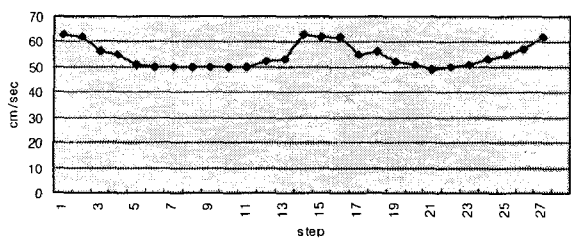


그림 11 전방 장애물이 없을 때의 주행경로
 Fig. 11 Navigation trajectory without a front obstacle

장애물은 복도 우측에 나와 있는 자연 장애물(책상, 휴지통)만 있으므로 전방에 장애물은 없는 것으로 인식된다. 그림에 점선은 기존 제어를 사용한 로봇의 주행 경로이고, 실선은 제안된 제어기 실험에서 로봇의 주행 경로를 나타낸다. 그림에 나와 있는 것처럼 두 제어기는 전방 장애물이 없는 경우에는 비슷한 경로를 그리면서 주행한다. 이러한 이유는 영상 센서에서 장애물 인식을 하지 못하거나, 복도 벽이 영상 센서에서 장애물로 인식될 때 복도의 방향 정보는 알 수 없으므로, 실제로 sonar 센서만 사용한 경우와 차이가 없는 결과를 가져온다. 그림 12는 제안된 제어기에서 계산된 로봇의 속도 정보와 각속도 정보를 그래프로 표시한 것이다.



(a) 제안된 제어기의 각속도 계산값



(b) 제안된 제어기의 속도 계산값

그림 12 제안된 제어기의 각속도 및 속도값
 Fig. 12 Angle velocity and velocity values of proposed controller

속도 변화와 각속도 변화를 살펴보면 각속도의 변화가 심할 때 속도가 줄어들면서 로봇의 주행 안전성을 높이는 특징을 살펴볼 수 있다. 또한 극단적인 상황을 맞지 않은 위와 같은 주행 상황에서는 속도의 변화를 억제하여 주행의 성능을 높인 것도 확인 할 수 있다.

(2) 전방 장애물이 있을 때의 주행

그림 13은 실험이 행해진 복도의 상황을 그림으로 나타낸 것이다.

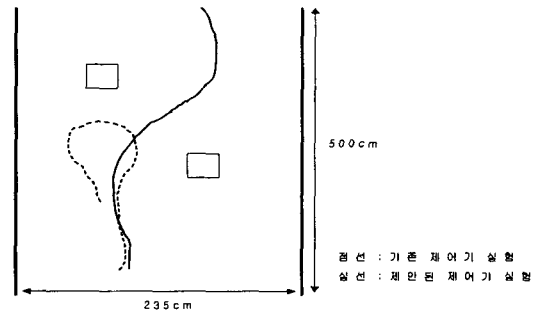
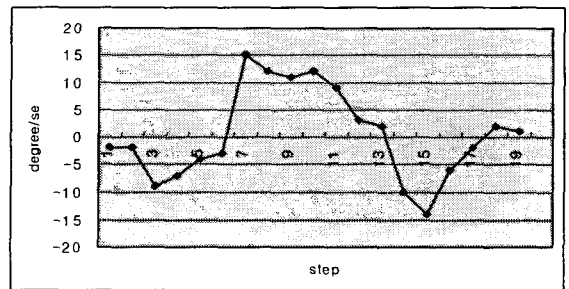
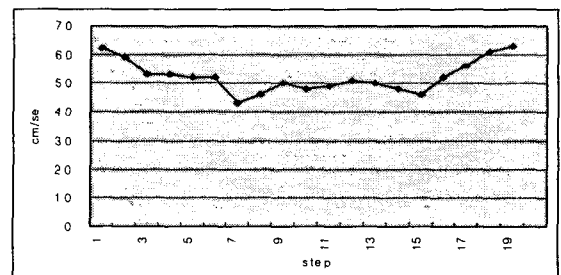


그림 13 전방 장애물이 있을 때의 주행경로
 Fig. 13 Navigation trajectory without a front obstacle

이번 실험에서는 로봇의 전방에 인공 장애물을 설치하여 기존 제어기와 제안된 제어기의 성능을 알아보았다. 제안된 제어기를 사용한 로봇은 장애물 사이의 공간을 인식하여 장애물을 회피하고 직진했지만, 기존 제어기를 사용한 로봇의 실험에서는 전방 장애물을 벽으로 인식해서 로봇이 출발지점으로 복귀하는 사태가 발생했다. 이번 실험을 통하여 제안된 제어기가 전방 장애물 인식에 유용한 수단이 될 수 있음을 입증하였다. 그림 14는 제안된 제어기에서 계산된 로봇의 속도와 각속도 변화를 그래프로 나타냈다.



(a) 제안된 제어기의 각속도 계산값



(b) 제안된 제어기의 속도 계산값

그림 14 제안된 제어기의 각속도 및 속도값
 Fig. 14 Angle velocity and velocity values of proposed controller

그림을 살펴보면 앞의 실험보다 평균속도가 떨어진 것을 알 수 있는데, 이유는 앞의 실험보다 각속도 변화가 심하여 로봇의 주행 안전성을 높이기 위하여 주행속도를 줄였기 때문이다.

3. 결 론

sonar 센서는 방사각에 대한 취약한 해석력을 가지고 있어서, 물체의 방향 탐지에 적합하지 않다. 비전 센서는 영상 외란으로 인하여 장애물 영상을 추출하기가 쉽지 않다. 본 논문에서는 sonar 센서의 정확한 거리정보와, 영상 센서의 풍부한 정보량을 조합하여, 장애물 위치 측정의 새로운 방법을 제안하였다. 분할/통합 알고리즘은 윤곽선검출 알고리즘보다 영상왜란에 강하고, 장애물을 단일 영역으로 만드는 과정이 효율적이다. 이를 확인하기 위해 실험을 통하여, 제안된 알고리즘으로 처리된 영상과 윤곽선 검출 영상을 비교하였다. 또한, sonar와 비전정보를 조합함으로써, 장애물의 거리와 방향 정보를 동시에 획득할 수 있었다.

영상 센서를 통해서 얻어진 장애물의 위치와 거리는 sonar 센서에서 얻은 복도 거리차 정보와 함께 퍼지 주행 제어기의 입력 변수로 사용된다. 본 논문에서는 퍼지 제어기를 설계하는데 있어서, 장애물을 회피하기 위한 각속도 제어기와 복도 중앙 주행을 위한 각속도 제어기를 분리해서 설계하여 주행 중에 전방 장애물을 만났을 때 로봇이 혼란에 빠질 위험성을 최소화하였다. 이를 위하여 제안된 제어기의 알고리즘을 시뮬레이션 해보았으며, 실제 실험을 통하여 안정성 및 효율성을 입증하였다. 향후 효율적인 장애물 인식을 위해 영상처리과정에서 통합과정을 위한 경계분할값 N을 적절한 값으로 선정하는 문제와 각 센서에서 주어진 정보를 토대로 map building을 작성하여 장애물 회피를 포함한 목표물까지의 최단거리를 계산하는 과정이 필요하다.

참 고 문 헌

[1] S. Hutchinson, G. D. Hager, and P. I. Corke, "A Tutorial on Visual Servoing Control", IEEE Trans. on Robotics and Automation, Vol. 12, No. 5, pp. 651-670, Oct, 1996.

[2] L. E. Weiss, A. C. Sanderson, and C. P. Neuman, "Dynamics sensor-based control of robots with visual feedback", IEEE J. Robot Automat., Vol. RA-3, No.5, pp. 404-417, Oct, 1987.

[3] W. J. Wilson, C. C. Jullis, and G. S. Bell, "Relative end-effector control using Cartesian position based visual servoing", IEEE Trans. on Robotics and Automation, Vol. 12, No. 5, pp. 684-696, Oct, 1996.

[4] K. Hashimoto, T. Kimoto, T. Ebine and H. Kimura, "Manipulator Control with Image-based Visual Servo", IEEE Int. Conf. on Robotics and Automation, pp. 2267-2272, April 1991.

[5] R. Kuc, and B. Barshan, "Docking Mobile Robots Using a Bat-like Sonar", IEEE/RSJ Int. Conf. on Intelligent Robots and Systems. Vol.2, pp. 1439-1444, July 1992.

[6] P. Mira Vaz, R. Ferreira, and V.I. Robeiro, "Docking of a mobile platform based on infrared sensors", Proceedings of the IEEE Int. Symposium of Industrial electronics. Vol. 2, pp. 735-740, 1997.

[7] R. C. Arkin, R. Murphy, M. Pearson and D. Vauhg, "Mobile Robot Docking Operation in a Manufacturing Environment: Progress in Visual Perceptual Strategies," IEEE/RSJ Int. on Robots and Systems, pp. 147-154, 1989.

[8] Charkari, M., Ishii, K., and Mori, H. "Proper selection of sonar and visual sensors for vehicle detection and avoidance", IROS, pp1110-1117, 1994.

[9] N. Matsui, "Navigation control of a mobile robot based on active vision", Proceedings. IECON '91, pp. 1122-1126 Nov. 1991.

[10] I. Pitas, "Digital Image Processing Algorithms and Applications", Wiley Interscience, 1997.

[11] Y. H. Joo, H. S. Hwang, K. B. Kim and K. B. Woo, "Fuzzy System Modeling and Its Application to Mobile Robot Control", Fuzzy Logic and Its Applications to Engineering, Information Sciences, Intelligent Systems, International Publication of Kluwer Academic Publishers, pp. 147-156, Nov. 1995.

저 자 소 개



김 정 훈 (金正勳)

1975년 4월 11일생. 2002년 건국대 전기공학과 졸업. 2004년 고려대 전기공학과 졸업(석사). 2004년~현재 LG전자 연구원



임 묘 택 (林畝澤)

1963년 3월 26일생. 1985년 고려대 전기학과 졸업. 1987년 동 대학원 전기공학과 졸업(석사). 1990년 미국 Rutgers Univ. 졸업(석사). 1994년 미국 Rutgers Univ. 졸업(박사) 1994년 7월~1995년 2월 삼성종합기술원 Mechatronics 연구실 선임연구원. 1995년 3월~1996년 8월 창원대학교 제어계측공학과 전임강사. 1996년 9월~현재 고려대학교 전기전자전파공학부 부교수



김 영 중 (金瑛中)

1973년 3월 14일생. 1999년 고려대 전기공학과 졸업. 2001년 동 대학원 전기공학과 졸업(석사). 2001년~현재 동 대학원 전기공학과 박사과정