

어안렌즈와 천장의 위치인식 마크를 활용한 청소로봇의 자기 위치 인식 기술

Location Identification Using an Fisheye Lens and Landmarks Placed on Ceiling in a Cleaning Robot

강 태 구, 이 재 현, 정 광 오, 조 덕 연, 임 충 혁, 김 동 환*

(Taegu Kang, Jae Hyun Lee, Kwang Oh Jung, Deok Yeon Cho, Choog Hyuk Yim, and Dong Hwan Kim)

Abstract: In this paper, a location identification for a cleaning robot using a camera shooting forward a room ceiling which has three point landmarks is introduced. These three points are made from a laser source which is placed on an auto charger. A fisheye lens covering almost 150 degrees is utilized and the image is transformed to a camera image grabber. The widely shot image has an inevitable distortion even if wide range is covered. This distortion is flattened using an image warping scheme. Several vision processing techniques such as an intersection extraction, erosion, and curve fitting are employed. Next, three point marks are identified and their correspondence is investigated. Through this image processing and image distortion adjustment, a robot location in a wide geometrical coverage is identified.

Keywords: fisheye lens, cleaning robot, image warping, landmarks, image processing

I. 서론

로봇 청소기와 그 시스템 및 제어 방법에 관한 것으로서 로봇 청소기가 주행하면서 촬상된 상방 이미지를 이용하여 주행을 조정하는 로봇 청소기이다.

로봇 청소기의 위치 인식하는 방식으로는 상방 카메라를 이용하여 벽면 형광등 액자 등 특징점을 잡아 로봇의 위치를 인식하는 방식이 있다[1,2]. 이러한 방식은 정확한 특징점을 잡는데 어려움을 나타내며 그로인해 위치인식을 하는데 어려움을 보인다. 또한 카메라를 이용하지 않고 로봇 본체에 설치된 초음파 센서를 이용하여 벽이나 장애물로 둘러싸인 청소영역의 외곽을 주행하면서 바퀴의 회전수 및 회전 각도를 검출하는 센서를 통해 검출된 신호로부터 주행거리 및 현재 위치를 산출하고 청소할 영역을 결정하고, 결정된 청소 영역을 청소경로를 계획한다[3,4]. 그런데 이러한 위치 인식 방식은 로봇 청소기가 주행하는 동안 바퀴의 슬립현상, 바닥면의 굴곡 등으로 인해 센서로 검출된 신호로부터 산출된 주행거리 및 이동 위치와 실제 주행거리 및 위치와의 오차가 발생한다. 이러한 오차가 누적될수록 정확한 위치인식이 되질 않아 청소기가 계획된 청소 경로를 이탈할 수 있다. 그 결과, 청소작업이 수행되지 않은 영역이 발생하거나, 동일 영역을 수회 반복 청소할 수 있어 작업 효율을 떨어뜨리는 문제점이 있다. 또 다른 방법은 RFID나 여러 유형의 인식마크를 붙여놓고 자기 위치를 인식하는

방법이 있지만 이 경우 많은 수의 마크를 설치해야 하고 갑작스런 공간의 변화에 대해 로봇이 인식 못 하는 한계가 있다[5-7].

위와 같은 문제점을 보완하기 위해 착안된 것으로서, 로봇 청소기의 현재 위치를 보다 정확하게 인식하고, 목적지로의 이동, 청소 주행 궤도를 보다 효율적으로 수행할 수 있는 로봇 청소기와 그 시스템 및 제어 방법을 제공하는데 본 연구의 목적을 두고 있다. 상기의 목적을 달성하기 위하여 로봇 청소기는 본체의 주행 방향에 수직인 상방의 이미지를 촬상할 수 있도록 본체상의 상방카메라(CCD)를 설치하였으며 또한 보다 넓은 영역을 확보하게 위하여 어안렌즈를 카메라에 부착하였다. 보다 효과적인 특징점을 확보하기 위하여 충전 스테이션에서 세 포인트의 레이저를 천장 방향으로 주사한다. 이 세 포인트를 이용하여 로봇 본체에 설치된 상방카메라(CCD)를 이용하여 세 포인트의 좌표정보를 인식하여 충전스테이션에서 로봇과의 거리, 현재 로봇의 위치, 방향을 정보를 알 수 있다. 어안렌즈에 의한 넓은 영역 이미지를 획득하는데 유리하지만 촬영된 이미지는 하나의 영상안에 보이기에 상당한 왜곡이 동반된다. 이러한 왜곡된 영상 이미지를 영상처리 기법을 통하여 온전한 영상으로 변화를 시켜야한다.

상기 인식용 세 포인트 데이터를 왜곡 영상으로부터 보정된 영상에서 그 위치값으로 환산하여 로봇의 현재 위치를 계산하게 된다.

또한, 상기의 목적을 달성하기 위하여 로봇 청소기 시스템은 복수의 바퀴를 구동하는 구동부와, 주행 방향의 수직인 상방의 이미지를 촬상 할 수 있는 영상부, 청소기의 작업영역의 천장에 인식용 세 포인트, 그리고 상방 카메라에 의해 촬상된 이미지정보를 이용하여 위치를 인식하고, 영상 왜곡 보정 알고리즘으로 구성된다.

* 책임저자(Corresponding Author)

논문접수: 2009. 4. 7., 수정: 2009. 5. 5., 채택확정: 2009. 8. 5.

강태구, 정광오, 조덕연: 서울산업대학교 기계설계자동화 공학부 (kangtg@paran.com/jko83@hotmail.com/cammel@nate.com)

이재현: 서울산업대학교 메카트로닉스학과(ljhpooh80@naver.com)

임충혁, 김동환: 서울산업대학교 기계설계자동화 공학부

(chyim@snut.ac.kr/dhkim@snut.ac.kr)

II. 청소로봇의 자기위치 인식 시스템

본 연구에서 개발된 지능형 청소로봇은 기존의 청소로봇이 가지고 있는 브러시를 이용한 청소 기능뿐만 아니라 청소모드 시 임의로 이동하는 무 방향성을 지양하고 자신의 위치 정보를 이용하여 청소를 효율적으로 하는 데 역점을 두고 있다. 그림 1은 개발된 청소로봇에 장착되어 있는 센서들을 표기하고 있다. PSD (Position Sensitive Device) 센서와 초음파센서는 전방에 대한 장애물에 대한 거리 정보를 제공하며 각 센서별 장단점을 활용하여 장애물 인식에 대한 정확도를 높이고자 하였다.

그림 2는 로봇의 자기 위치를 인식하기 위한 센서 시스템을 보여준다. 여기서 고정부(home base)에서 레이저를 실내 청정에 비추어 주변 천정에는 3개의 원형 표식이 나타난다. 이 3개의 표식을 로봇 카메라로부터 읽어 이미지처리를 하여 현재의 로봇의 위치를 추정하게 된다. 특히 방의 크기가 8-10m 까지 표식의 이미지를 찾기 위하여 넓은 각도로 이미지를 획득하는 어안렌즈를 채용하였다. 이 렌즈는 150도 정도의 넓은 각도로 이미지를 보여주지만 이미지 외곽선에서는 상당한 정도의 이미지의 왜곡이 발생하게 된다. 이러한 이미지는 특별한 이미지 처리를 통하여 왜곡된 부분을 편평한 이미지로 복원하여 3점의 위치에 대한 정확한 위치를 재계산하게 된다.

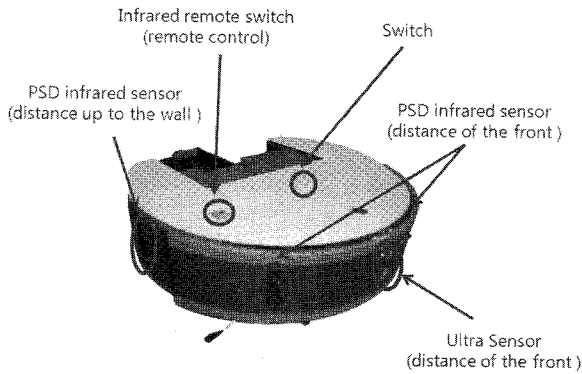


그림 1. 개발된 청소로봇에 적용된 센서.
Fig. 1. Sensors installed on the developed cleaning robot.

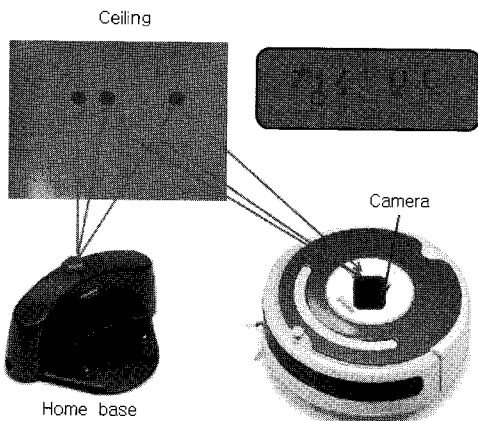


그림 2. 자기위치를 인식하기 위한 시스템.
Fig. 2. Self location identification system.

III. 영상처리 과정

로봇의 자기위치 인식을 위한 영상처리 과정을 그림 3에 표시하였다. 어안렌즈[8]를 사용하기 때문에 표준영상을 획득하면 이미지왜곡이 나타나게 된다. 이에 왜곡 보정하여 영상을 추출하고 레이저의 적색을 검출하고 필터링 과정을 거친 후 라벨링 작업을 하여 각각 포인트를 검출한다. 천장 높이를 판단하게 되면 현재 로봇의 위치인식 (로봇의 충전 스테이션과 상대적 거리 및 방향, 로봇의 방향)을 판단한다. 벽 추적(wall following) 신호가 들어오면 포인트가 검출되는 영역의 맵을 획득한다.

1. 영상 왜곡 보정 관련 연구

광각 렌즈에 의해 전송된 영상에는 심한 왜곡이 존재한다. 왜곡은 일반적으로 피사체의 직선이 영상에서는 곡선 형태로 나타난다. 이러한 렌즈의 왜곡을 보정하기 위한 기존의 연구들은 왜곡 계수를 찾기 위하여 실제계상의 알고 있는 점들을 이용하는 방법[8], Calibration Object를 사용하는 방법[13], 사용자 하여금 영상으로부터 이용할 Object를 선택하는 방법[11]들로 모두가 기존에 제안된 아래와 같은 복잡한 왜곡 모델을 기반으로 한다. 그림 4에서 피사체의 한 점(q)은 투시사영(perspective project)과 렌즈의 의해 영상에서는 q'으로 나타난다.

$C(x_p, y_p)$: optical center 라 할 때 Cartesian과 polar 좌표는 식 (1)과 같이 나타낼 수 있다.

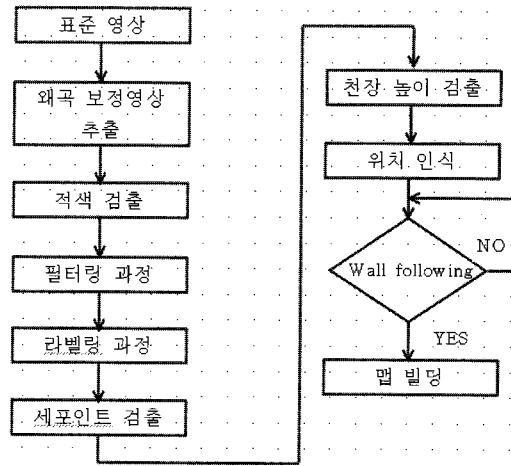


그림 3. 로봇의 자기위치 인식을 위한 이미지 처리 과정.
Fig. 3. Image processing procedures for identifying robot location.

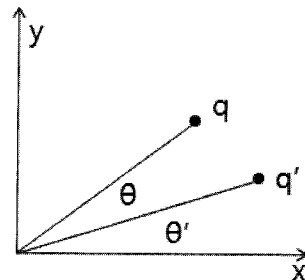


그림 4. 한 점의 영상 투시 사영(q가 점 q'으로 매칭).
Fig. 4. Image projection for one point (q corresponds to q').

$$r = \sqrt{\bar{x}^2 + \bar{y}^2}, \tan(\theta) = \frac{\bar{y}}{\bar{x}}, (\bar{x} = x - x_p, \bar{y} = y - y_p) \quad (1)$$

q의 왜곡은 영상중심의 이동(x_p, y_p), 방사왜곡, 이탈왜곡요소나 나누어진다. 방사왜곡의 거의 모든 광각렌즈에서 일어나며 광학적 중심을 향하여 점들이 끌려지는 형태로 대칭적 방사의 모양을 갖게 된다. 방사왜곡 모델의 왜곡 계수는, 차수가 5 이상이면 왜곡에 영향이 적으므로 아래의 식으로 표현된다.

$$\Delta r(q) \approx C_3 r^3 + C_5 r^5 \quad (2)$$

중심 이탈 왜곡은 광축에 관계되는 렌즈요소의 비직교성이 원인이다. 이것은 모델은 식 (3)로 나타낼 수 있다. 여기서 P_i 는 왜곡계수이며, $\Delta T_x, \Delta T_y$ 는 x, y 방향에 대한 왜곡이다.

$$\begin{aligned} \Delta T_x(q) &\approx [P_1 r^2(1 + 2\cos^2(\theta)) + 2P_2 r^2 \sin(\theta)\cos(\theta)] \\ \Delta T_y(q) &\approx [P_2 r^2(1 + 2\sin^2(\theta)) + 2P_1 r^2 \sin(\theta)\cos(\theta)] \end{aligned} \quad (3)$$

전체적인 왜곡은 식 (2)와 식 (3)을 더한 것으로 식 (4)로 나타낼 수 있다.

$$\begin{aligned} \Delta x(q) &\approx \cos(\theta)[\Delta r(q)] + \Delta T_x(q) \\ \Delta y(q) &\approx \sin(\theta)[\Delta r(q)] + \Delta T_y(q) \end{aligned} \quad (4)$$

그러므로 왜곡을 보정하기 위해서는 계수 C_3, C_5, P_1, P_2 를 얻는 것이 필요하다. 기존에 제한된 방법들은 이러한 보정계수를 구하기 위하여 직선으로 간주되지만 곡선으로 나타난 영역의 점들을 사용하여 그 영역을 직선으로 근사화(fitting)하기 위하여 그 직선과 수평축이 만나는 각을 θ , 영상의 중심으로부터 직선까지의 거리를 ρ 라 하면 왜곡되지 않은 하나의 점 q 에 관계되는 에러(e)를 식 (5)로 나타낼 수 있다.

$$e = (x' \sin(\theta) - y' \cos(\theta) + \rho)^2 \quad (5)$$

직선이 곡선으로 변화된 영역에 속한 점들에 대하여 e가 최소가 될 때의 파라메타(C_3, C_5, P_1, P_2)를 구한다. 이러한 과정은 복잡하며 일일이 직선이 곡선으로 변화된 영역을 사용자가 정의 하거나 실세계의 직선을 이루는 점들의 좌표가 미리 알려져 있어야만 구할 수 있었다. 그러나 본 논문에서는 표준 영상과 왜곡 영상 간에 매핑되는 그리드 교차점을 찾아 이들 간의 관계를 3차식으로 모델링 하는 방안을 채용하여 보정계수를 구하여 왜곡 영상을 개선한다. 이 방안은 기존에 알려진 방법이며 그 상세 내용은 참고문헌 [9]에 잘 명시되어 있으므로 상세 과정은 생략한다. 단지 마지막 단계에서 3*3 마스크를 채용하여 교차점 추출의 정확성을 높였다.

2. 왜곡 보정 영상 추출 알고리즘

그림 5는 왜곡 영상을 자동 보정하기 위한 알고리즘의 흐름도를 나타내고 있다. 또한 왜곡 영상 격자의 교차점들을 구하기 위한 과정도 나타내었다. 교차점들을 구하기 위

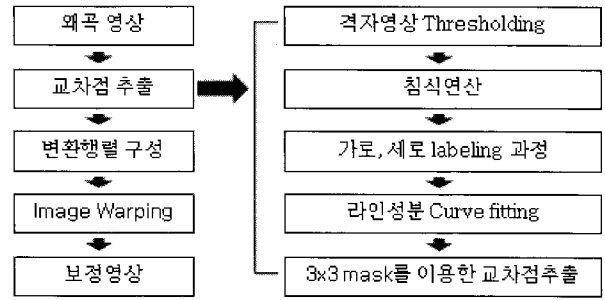


그림 5. 왜곡 보정 알고리즘.

Fig. 5. Compensation algorithm of image distortion.

표 1. 왜곡 영상과 보정 영상에 대한 통계 (단위: 픽셀).

Table 1. Statistics of distorted image and compensated image.

	평균오차 (E_{avg})	최대오차 (E_{max})	표준편차 (S_D)	Error 복원률 (%) (E_{re})
왜곡 영상	8.3452	33.7788	6.2334	0
보정 영상	0.2425	1.2341	0.1745	97.2314

해서는 수직, 수평 방향에 대하여 각 과정을 각각 수행한 후 3x3 마스크(mask)를 이용하여 교차점을 추출한다. 이렇게 해서 구해진 왜곡 영상에서의 교차점들과 표준 영상에서의 교차점들의 집합으로부터 이 두 집합사이를 변환시켜 주는 변환 행렬을 구한다. 구해진 변환 행렬을 이용하여 왜곡된 영상을 영상워핑 시킴으로써 보정 영상을 얻는다.

3. 왜곡 영상 처리 결과

영상왜곡 처리 과정을 수행한 후 표 1에 보정 결과에 대한 통계 자료를 실었다. 오차의 측정은 표준 영상 격자의 교차점들과 매칭 되는 각 영상 격자의 교차점들 사이의 거리에서 구해진다.

표 1에 제시된 성능지수는 아래에서 정의한 식을 사용하였다.

$$\begin{aligned} E_{avg} &= \left(\sum_1^N d_i \right) / N \\ E_{max} &= \text{Max}(d_i) \quad i = 1, 2, \dots, N \\ S_D &= \left(\sum_1^N (d_i - E_{avg})^2 \right) / N \\ E_{re} &= 100 - (E_{avg1} \times 100) / E_{avg2} \end{aligned} \quad (6)$$

식 (6)에서 d_i 는 표준 영상 격자의 교차점들과 매칭 되는 각 영상 격자의 교차점들 사이의 거리를 나타내고 N 은 교차점의 수를 나타낸다. E_{avg1} 은 보정 영상의 평균 오차를 나타내고 E_{avg2} 는 왜곡 영상의 평균 오차를 나타낸다.

표 2에 distortion 에 대하여 식 (3)에서 쓰인 변환 행렬의 각 인자값들 $a_0 \sim a_9, b_0 \sim b_9$ 를 나타내었다. 그림 6은 위에서 구한 계수를 사용하여 이미지 워핑을 통한 보정된 이미지를 보여주고 있다.

표 2. 변환 행렬에 쓰는 각 계수 값들.

Table 2. Coefficients used for transformation matrices.

계수	값	계수	값
a_0	33.77885207667464	b_0	23.40560173758564
a_1	0.51094697249524	b_1	-0.19096349178874
a_2	-0.15364722318311	b_2	0.67625712898576
a_3	0.00096653107477	b_3	0.00160323387906
a_4	0.00264370192600	b_4	0.00060067281251
a_5	0.00064450750870	b_5	0.00167154524723
a_6	-0.00000000000385	b_6	-0.00000503566608
a_7	-0.00000405432944	b_7	-0.00000000076544
a_8	-0.00000554545448	b_8	-0.00000000270072
a_9	0.00000000000185	b_9	-0.00000473979904

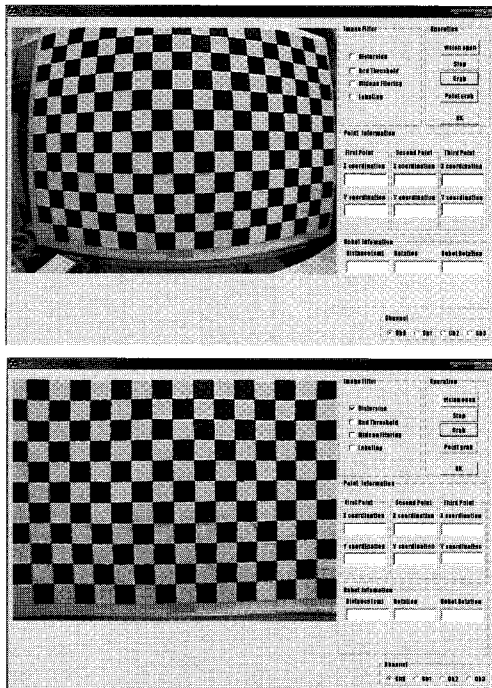


그림 6. 왜곡 보정 전(상)과 보정 후(하)의 이미지.
Fig. 6. Images before image compensation (top) and after image compensation (bottom).

IV. 포인트를 이용한 로봇의 위치 파악

1. 포인트 추출 알고리즘

세 포인트에 대한 값을 추출하기 위해서는 먼저 왜곡 보정된 영상으로부터 포인트의 색인 적색을 판별하기 위해 적색 값을 경계(thresholding)하여 최대한 노이즈를 없애준다. 또한 이러한 노이즈 값을 없애 주기 위해 먼저 카메라 자체의 부분인 밝기 값과 경계 값을 조절 할 수 있게 하였으며 이 값을 이용하여 최대한 적색의 이미지만 남게 한다. 다음 미디언 필터링(low pass) 을 거쳐 한 번 더 노이즈를 제거 한다. 노이즈의 제거가 완료되면 각 포인트에 대한 라벨링을 거쳐 각각의 포인트의 크기와 개수, 좌표

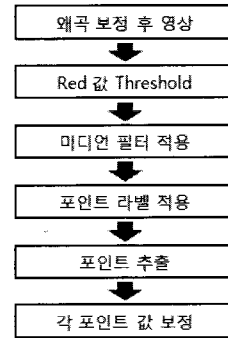


그림 7. 포인트 추출 알고리즘.
Fig. 7. Algorithm for extracting points.

값을 산출한다. 이 부분에서 노이즈와 포인트를 구분하기 위해 크기에 대한 경계 값을 지정하였으며, 또한 산출된 포인트의 개수가 지정된 개수이상이면 들어온 이미지 한 프레임 자체를 연산을 하지 않는 방법으로 노이즈를 구분 하였다(그림 7).

2. 천장 높이와 각 포인트를 이용한 로봇의 위치 파악

레이저 포인트의 직진성과 천장의 표면이 바닥과 수평을 이루는 것을 가정으로 천장높이를 검출하였다. 먼저 그림 13에서 알 수 있듯이 천장 높이가 높아질수록 카메라 이미지의 포인트의 위치가 중심으로부터 거리가 멀어 지는 것을 알 수 있다. 이를 바탕으로 바닥과 레이저 포인트의 거리=20cm, $\theta_3 = 30^\circ$, θ_1 과 θ_2 는 천장의 높이가 250cm로 가정하여 p1과 p2의 이격거리 10cm, p1과 p3의 이격거리 20cm로 정한 후 각도를 결정하였다. 그 결과 $\theta_1 = 1.25^\circ$, $\theta_2 = 2.50^\circ$ 로 결정하였다.

표 3은 $\theta_1, \theta_2, \theta_3$ 의 각도와 p'1과 p'2, p'3의 거리차(pixel)을 이용하여 높이 검출하였다. 보정된 이미지를 이용하여 실제 천장을 볼 수 있는 넓이를 이용하여 1pixel 당 0.8cm 단위를 알 수 있다. 이를 바탕으로 각 포인트의 좌표점을 이용하여 높이를 검출 하였다 (그림 8).

$$d_1 = (p'1 \text{과 } p'2 \text{와의 이격거리} (\pi \text{xel}) * 1.25) / \tan \theta_1$$

$$d_2 = (p'1 \text{과 } p'3 \text{와의 이격거리} (\pi \text{xel}) * 1.25) / \tan \theta_2$$

$$d = (d_1 + d_2) / 2 \quad (d_1 \text{과 } d_2 \text{의 평균값}) \quad (7)$$

$$h(\text{cm}) = 20 + d \sin \theta_3$$

식 (7)을 이용하여 검출한 높이는 각 포인트의 좌표점을

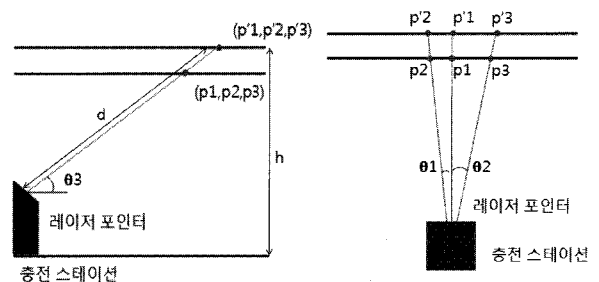


그림 8. 레이저 포인트를 이용한 천장높이 검출 방법.
Fig. 8. Ceiling height measurement using a laser point.

표 3. 거리차(pixel)을 이용하여 높이 검출(단위 환산 1pixel=약 0.8cm).

Table 3. Ceiling Height measurement using pixel (1pixel=0.8 cm).

p1과 p2의 거리 차(pixel)	p1과 p3의 거리 차(pixel)	높이 검출(cm)	실제 높이(cm)	오차(cm)
8	16	250	250	0
10	20	263	260	3
13	25	275	270	5
15	30	283	280	3
17	33	295	290	5
19	36	308	300	8

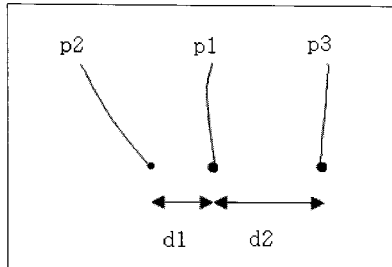


그림 9. 어안렌즈의 상방 카메라 촬상 이미지.
Fig. 9. Three points viewed from camera with fisheye lens.

추출한 후 로봇의 위치를 결정하는 요소인 c에 적용된다. 세 점의 좌표 값을 산출하여 각각의 영상 데이터인 (x_1, y_1) , (x_2, y_2) , (x_3, y_3) 값을 이용하여 로봇의 위치 파악한다(그림 9).

그림에서 p1 : 기준 포인트이며 p2와 p3은 로봇의 방향을 알 수 있게 형성된 포인트이고 d1과 d2는 각각 p1과 p2의 거리와 p1과 p3의 거리를 나타낸다.

로봇 청소기의 충전 스테이션과의 상대적 거리 값과 방향을 p1의 좌표를 통해 획득하며 p2와 p3를 이용하여 로봇의 방향 정보를 획득한다. 각 포인트 추출로 인해 중심점에서 각 거리값 비교 하여 기준이 되는 포인트를 이용하여 로봇의 각도를 산출한다.

$$d1 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{8}$$

$$d2 = \sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2}$$

($d1 > d2$) 일 경우 (로봇의 각도 180~360)

$$rotation = \tan^{-1}[(y_1 - y_2)/(x_1 - x_2)] + 360 \tag{9}$$

($d1 < d2$) 일 경우 (로봇의 각도 0~180)

$$rotation = \tan^{-1}[(y_1 - y_3)/(x_1 - x_3)] \tag{10}$$

로봇의 거리 값 산출은

$$distance = \sqrt{x_2^2 + y_2^2} * c \tag{11}$$

(천장의 높이 따른 변수 값 $c=0.46875$)

산출된 로봇의 회전각도 값을 이용하여 이미지상의 포인트를 값을 적용하여 로봇의 위치를 파악한다. 이미지 중심점을 기준으로 정하고 회전 값을 이용하여 보정 (x, y) 값을 구한다.

$$\begin{aligned} x &= x_2 * \cos(rotation * \pi / 180) \\ &+ y_2 * \sin(rotation * \pi / 180) + 0.5 \\ y &= -x_2 * \sin(rotation * \pi / 180) \\ &+ y_2 * \cos(rotation * \pi / 180) + 0.5 \end{aligned} \tag{12}$$

최종 (x, y) 에 0.5를 더하여 반올림을 하여 (x, y) 값을 보정하였다. 또한 로봇의 카메라의 위치가 로봇의 중심에 있지 않고 15cm 벗어난 곳인 로봇의 상부에 부착되어 이에 따른 변경사항을 적용함으로써 실 좌표 값인 (x, y) 로 다시 변경하였다. 위의 식을 이용하여 로봇의 위치를 파악 할 수 있다.

$0 < rotation < 90$ 인 경우

$$\begin{aligned} x &= x + robotR * \sin(rotation * \pi / 180) \\ y &= y - roboyR * \cos(rotation * \pi / 180) \end{aligned} \tag{13}$$

(robotR=15cm)

$90 < rotation < 180$ 인 경우

$$\begin{aligned} x &= x + robotR * \sin(rotation - 90 * \pi / 180) \\ y &= y - roboyR * \cos(rotation - 90 * \pi / 180) \end{aligned} \tag{14}$$

$180 < rotation < 270$ 인 경우

$$\begin{aligned} x &= x + robotR * \sin(rotation - 180 * \pi / 180) \\ y &= y - roboyR * \cos(rotation - 180 * \pi / 180) \end{aligned} \tag{15}$$

$270 < rotation < 360$ 인 경우

$$\begin{aligned} x &= x + robotR * \sin(rotation - 270 * \pi / 180) \\ y &= y - roboyR * \cos(rotation - 270 * \pi / 180) \end{aligned} \tag{16}$$

여기서 오른쪽 향의 (x, y) 는 카메라 위치를 고려하기 전의 좌표 값이며, 왼쪽 향의 (x, y) 는 로봇의 카메라 위치를 고려하여 보정된 좌표 값이다. 그림 10은 실제 촬영한 세 인식마크의 영상을 이미지 워핑을 거쳐 보정된 이미지를 구하고 각 인식마크가 놓여진 위치를 기준으로 현재 로봇의 위치 및 방향을 계산하는 프로그램이다.

3. 실 좌표를 이용한 영상 데이터

왜곡 보정 전 좌표 값의 데이터 값과 왜곡 보정후의 좌표 값을 산출하여 이를 바탕으로 왜곡 보정에 따른 정확도

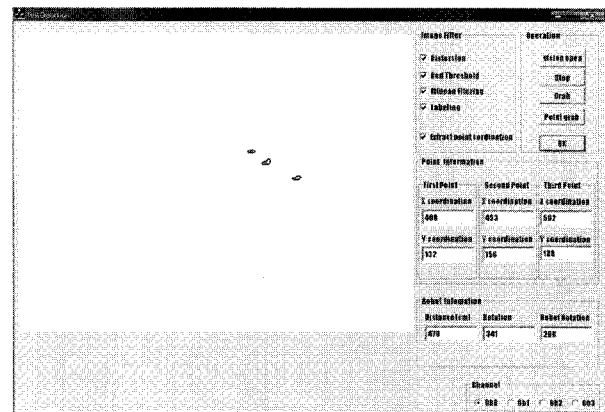


그림 10. 포인트 검출 및 위치인식.
Fig. 10. Program for detection and identification of three points.

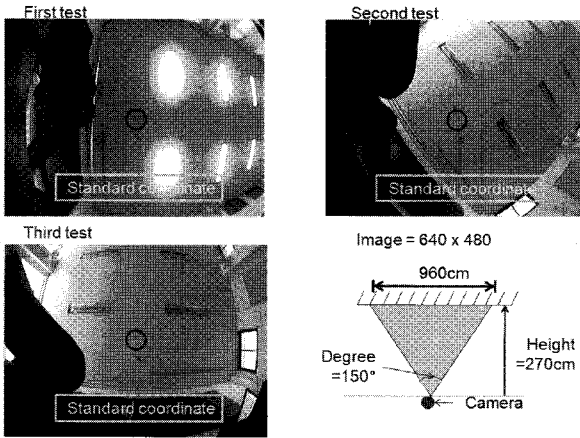


그림 11. 데이터 추출 실험 환경.

Fig. 11. Experiment environment for extracting data.

를 검출할 수 있습니다. 데이터 추출 방법으로 천장의 임의에 특징 점을 표시한 후 이에 대해 레이저 포인터를 이용하여 표시의 점에 적색을 나타나게 한 포인트를 추출 하였다. 이를 바탕으로 데이터를 추출하였다. 그림 11은 현재 실험 대상으로 설정한 공간을 나타낸다. 각 3가지 경우에 대해 실험 어안렌즈를 통하여 비쳐진 영상을 보여주고 있다. 어안렌즈는 150도 각도까지 영상을 획득할 수 있기에 빠른 영상의 정보 처리가 가능하다.

표 3에 나온 결과에 따르면 실험 데이터의 정확도는 왜곡 보정 전에는 최대 94%에서 최소 57%의 값을 나타냈다. 이 실험에서 이미지의 중심에서 거리가 멀어질수록 오차값

표 4. 거리에 따른 왜곡 보정전의 데이터 값을 이용한 정확도 테스트(이미지의 중심을 기준으로 거리 값을 측정).

Table 4. Distance measurement accuracy data before image distortion for various distance (measured based on image center).

Standard coordinate (x,y)	x	y	True distance(cm)	Measure distance(cm)	Error (cm)	Accuracy (%)
2, 2	92	3	90	135	45	50.00
	171	2	180	253	73	58.44
	232	2	270	348	78	71.11
	266	2	360	396	36	90.00
	312	3	450	468	18	96.00
5, 4	7	-93	90	117	27	70.00
	6	-168	180	206	26	85.55
	6	-230	270	282	12	95.74

Standard coordinate (x,y)	x	y	True distance(cm)	Measure distance(cm)	Error (cm)	Accuracy (%)
3, -1	78	-55	90	129	39	56.87
	152	-108	180	256	76	57.78
	205	-147	270	350	80	70.37
	232	-158	360	402	42	88.33
	250	-176	450	427	23	94.88
	265	-190	540	453	87	83.89
	278	-198	630	478	152	75.87
	285	-201	720	487	233	67.64

표 5. 거리에 따른 왜곡 보정전의 데이터 값을 이용한 정확도 테스트(이미지의 중심점을 기준으로 측정, 최대 실험 측정거리=900cm).

Table 5. Distance measurement accuracy data before image distortion for various distance (measured based on image center, maximum distance=900cm).

Standard coordinate (x,y)	x	y	True distance(cm)	Measure distance (cm)	Error(cm)	Accuracy(%)
0, 0	85	2	70	71	1	98.8915
	128	2	140	139	1	99.2857
	186	2	210	210	0	100.0000
	268	2	280	277	3	98.9285
	318	1	350	348	2	98.8571
	-86	1	70	72	2	97.2222
	-180	2	140	141	1	98.2857
	-182	2	210	209	1	98.6288
	-280	2	280	278	2	98.2857
	-318	2	350	348	2	98.8571
	88	50	90	95	5	98.7741
	190	86	180	185	5	98.6006
	184	145	270	274	4	98.5401
	288	184	360	362	2	99.4476
	319	259	450	454	4	98.5577
	2	50	90	91	1	98.8888
	2	98	120	120	0	100.0000
	1	144	160	160	0	100.0000
	2	194	240	242	2	98.1755
	2	288	300	305	5	98.0000

이 현저히 늘어났으며 (표 4), 왜곡을 보정한 이미지에서는 최대 100%에서 최소 97.22%를 나타냈다 (표 5). 오차 값이 이미지의 중심에서 멀어질수록 약간의 오차가 났다. 또한 이를 바탕으로 보정된 이미지로부터 포인터를 추출하였으므로 약간의 오차가 발생할 수 있다. 이러한 오차를 확인하고 이를 실험 데이터를 이용하여 거리에 따른 가중치를 주어 로봇의 위치와 거리 값을 보정한다.

4. 포인트 미 추출 시 보정 방법

4.1 검출된 포인트의 개수가 1, 2개일 경우

마지막 세 포인트 검출시의 회전 값을 기준으로 좌우 바퀴의 엔코더 값을 이용하여 회전 값을 보정한다. 이후 검출되는 포인트로 이동거리를 판단하여 로봇의 위치와 회전각을 산출한다.

4.2 검출된 포인트의 개수가 없을 경우

마지막 포인트 검출시의 회전각과 로봇의 위치를 기준으로 좌우 바퀴의 엔코더 값을 이용하여 이동거리와 회전각을 보정한 후, 로봇의 위치와 회전각을 산출한다.

4.3 포인트가 재 검출될 경우

최종 세 포인트가 검출 될 경우 정확한 로봇의 위치와 회전각을 좌우 바퀴의 엔코더 값을 이용하여 보정된 위치와 회전각과 을 비교하여, 세 포인트가 검출되지 않는 경우의 로봇의 이동 경로를 보정한다.

좌우 바퀴의 엔코더 값을 이용한 이동거리와 회전각, 카메라와 좌우바퀴의 엔코더 값을 혼용하여 측정된 이동거리와 로봇의 회전각의 오차 비교를 분석하였다. 표 6을 통해 알 수 있듯이 좌우 엔코더 값만을 이용한 로봇의 이동거리와 로봇의 회전각은 실제 이동거리와 회전각과의 오차는 최종측정거리인 10m를 이동하였을 경우 최대 오차는 이동거리 5.42%, 회전각 3.33%를 나타내는 반면, 좌우 바퀴의

표 6. 거리에 따른 오차율(A=좌우 바퀴의 엔코더 값을 이용한 경우, B=카메라와 좌우 바퀴의 엔코더 값을 이용한 경우).

Table 6. Errors for travelling distance (A=left/right encoders only, B=camera and left/right encoders).

이동거리(cm)	100	200	300	400	500	600	700	800	900	1000
A(cm)	102	205	310	418	525	630	738	842	948	1053
A오차율(%)	2	2.5	3.34	4.5	5	5	5.42	5.25	5.34	5.3
B(cm)	102	201	302	402	505	612	705	803	904	1006
B오차율(%)	2	0.5	0.67	0.5	1	2	0.71	0.38	0.44	0.6

기 존 각 도 (degree)	0	0	0	0	0	0	0	0	0	0
A(degree)	356	1	3	359	356	353	351	350	350	348
A오차율(%)	1.11	0.28	0.83	0.28	1.11	1.94	2.5	2.78	2.78	3.33
B(degree)	0	1	1	0	2	359	1	0	1	2
B오차율(%)	0	0.28	0.28	0	0.55	0.28	0.28	0	0.28	0.55

*장애물(탁자)의 위치는 이동거리 400-600cm 에 위치함.

엔코더 값과 카메라를 이용한 경우의 최대 오차는 이동거리 2%, 회전각 0.55%를 나타낸다. 따라서 카메라를 이용하여 천장의 세 포인트가 안보일 경우, 좌우 바퀴의 엔코더 값을 이용하여 보정하며, 세 포인트가 보이는 지역에서는 카메라를 이용하여 위치인식을 하는 방법이 효과적이다.

5. 기존 상방카메라를 이용한 방법과 비교 분석

참고문헌 [1]의 논문의 로봇의 위치인식의 단점은 상방 카메라와 passive landmark 방식을 이용하여 좁은 인식 영역을 확대하기 위해 어안렌즈를 사용하였지만, landmark의 6 m² 의 영역 내에서 이동로봇의 실험을 통하여 자기 위치 및 방위각을 인식하였습니다. 또한 발광 소자의 landmark가 아니므로 조명에 민감한 카메라의 특성으로 인해 정확한 인식이 되지 않는 문제점을 볼 수 있다. 이로 인해 실시간 인식의 오류로 인해 로봇의 이동경로를 파악하기 어렵다. 하지만 본 논문에서는 레이저 포인트를 이용하여 조명에 영향을 받지 않으며 카메라를 이용한 위치인식에서 좌표점이 획득 되지 않을 시 좌우바퀴의 엔코더 값을 이용하여 로봇의 이동거리와 회전각을 보정하였다. 기존 상방 카메라를 이용한 영상 데이터는 본 논문과 같은 어안렌즈를 이용하여 왜곡을 보정한 후 landmark를 추출하여 위치 데이터를 획득하지만, 본 논문은 왜곡 보정하는 프로그램을 별도로 두며, 이를 이용하여 획득된 왜곡 상수만을 사용하여 이미지 왜곡을 보정한다. 이로 인해 왜곡보정의 속도가 빨라졌으며, 또한 위치인식에 용이한 레이저 포인트를 인식하므로, 기존의 상방카메라를 이용하여 landmark, 방안의 형광등, 액자와 같은 특징점을 추출하여 위치를 획득하는 방법보다 속도가 빠르다.

상방 카메라 영상의 획득속도는 30 Hz 이나 왜곡 보정 후 landmark를 이용한 영상 획득 데이터 속도는 2~3 Hz로

느려지는 것으로 발표되었다. 다른 방안의 경우 전체 영상에서 landmark가 인식되면 로봇의 좌우 바퀴에 부착된 encoder sensor 값을 이용하여 landmark를 추종하는 방식은 20 Hz의 속도를 가지는 것으로 알려져 있다. 본 논문의 실험결과 왜곡 보정 후 레이저 포인트를 이용한 위치인식 속도가 27~28 Hz 정도로 상당히 빨리 처리되는 것으로 확인하였다.

본 논문은 왜곡 상수를 구하는 프로그램을 별도로 두어, 왜곡 보정하는 속도가 빠르며, 또한 위치인식의 속도 면에서도 기존의 논문인 왜곡보정 후 위치인식에 월등한 속도를 나타낸다. 또한 좌우 바퀴의 encoder 센서 값을 이용하여 landmark를 추종하는 방식 보다 빠른 속도를 나타낸다.

V. 결론

어안렌즈를 로봇 상체에 부착하여 천장에 있는 표식을 인식하여 로봇의 현재위치를 구하는 방식을 제안하였다. 여기서 적용한 기법을 통하여 별도의 센서 장치의 부착을 배제하여 비교적 손쉽게 로봇의 위치를 알 수 있다. 어안렌즈는 넓은 영역을 촬영할 수 있기에 하나의 카메라 이미지에 넓은 공간을 찍을 수 있으므로 위치 정보 처리가 빠르다. 그러나 이 찍힌 이미지는 상당한 왜곡이 발생한다. 이러한 왜곡은 이미지 워핑 기법을 사용하여 평탄한 이미지를 재생하고 재생된 이미지에 나타난 3개의 표식을 평탄한 영상에서 이미지 처리하여 로봇의 자기 위치 및 각도를 계산해 낼 수 있었다. 이러한 방법으로 5% 이내의 오차율을 가지는 상당히 정확한 위치 데이터를 추정 할 수 있었다. 이러한 방법은 기존의 상방 카메라를 이용한 미리 촬영한 영상 데이터에 의한 이미지 인식에 의한 위치 인식에 의한 방법보다 빠르고 장애물 특히 탁자 등에 의해 로봇 카메라가 가려졌을 경우에도 자기 위치 인식에 비교적 효과적으로 사용될 수 있는 방법이다.

참고문헌

- [1] 이주상, 임영철, 유영재, “유비쿼터스 이동로봇용 천장 인공표식을 이용한 비전기반 자기위치인식법,” 한국표지 및 지능 시스템학회 논문지, 제15권, pp. 560-566, 2005.
- [2] S. Wang, H. Chen, M. Strzelecki, 김형석, “천정 부착 칼라 패치 배열을 이용한 이동로봇의 자기위치 인식,” 제어·로봇·시스템학회 논문지, 제14권 제2호, pp. 156-160, 2008.
- [3] 손창우, 이승희, 이민철, “전자 나침반과 적외선 광원 추적을 이용한 이동로봇용 위치 인식 시스템,” 제어·로봇·시스템학회 논문지, 제14권 제8호 pp. 767-773, 2008.
- [4] Q. Zhang and R. Pless, “Extrinsic calibration of a camera and laser range finder (improves camera calibration),” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004.
- [5] M. Betke and L. Gurvits, “Mobile robot localization

using landmarks," *IEEE Transactions on Robotics and Automation*, vol. 13, no 2, pp. 251-263, 1997.

[6] G. Jang, S. Kim, W. Lee, and I. Kweon, "Color landmark based self-localization for indoor mobile robots," *ICRA2002*, Washington D.C., USA, May 2002.

[7] 최병석, 이장명, "RFID 센서 공간에서의 모바일 로봇의 효율적인 위치 인식," 제어 · 로봇 · 시스템학회 논문지, 제12권 제1호, pp. 15-22 2006.

[8] 강진아, 박재민, 김병국, "어안렌즈 왜곡보정에 관한 연구," 한국과학기술연구원 논문지, pp. 133-138, 2007.

[9] 이광진, "An auto correction method for lens distorted image," Hanyang Univ, 석사학위논문, 1999.

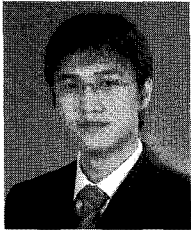
[10] 유원일, 정연구, "내장형 렌즈 왜곡보정 알고리즘 구

현을 위한 이미지 워핑 방법," 한국정보처리학회논문지, 제B권 10B호, pp. 373-380, 2003.

[11] 한기태, 김희율, "복합 왜곡영상을 위한 보정계수 자동추출 방법," 한국통신학회논문지, 제26권 3B호, pp. 302-314, 2001.

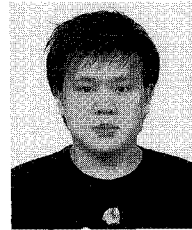
[12] R. Swaminathan and S. K. Nayar, "Non-metric calibration of wide-angle lenses and polycameras," *Proceeding Computer Vision and Pattern Recognition'99*, IEEE Computer Society Press, pp. 413-419, 1999.

[13] 송광열, 윤팔주, 이준웅, "어안렌즈왜곡 및 원근왜곡의 보정" 한국정밀공학회지, 제23권 제10호, pp. 22-29, 2006.



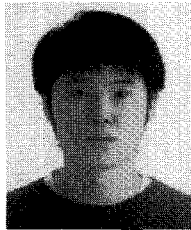
강 태 구

2009년 8월 서울산업대학교 기계설계 자동화공학부 졸업예정. 관심분야는 스테레오비전, 로봇틱스, 자동제어 및 최적제어, PC프로그램.



이 재 현

2007년 서울산업대학교 기계설계자동화공학부 졸업. 현재 서울산업대학교 메카트로닉스 석사과정. 관심분야는 메카트로닉스, 로봇틱스, 자동제어.



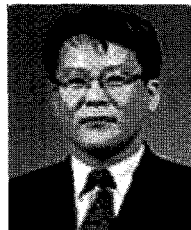
정 광 오

2009년 2월 서울산업대학교 기계설계 자동화공학부 졸업예정. 관심분야는 마이크로 로봇틱스, 자동제어 및 최적제어, 설계.



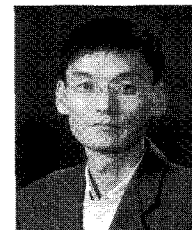
조 덕 연

2009년 8월 서울산업대학교 기계설계 자동화공학부 졸업예정. 관심분야는 로봇틱스, 기계설계, 자동제어 및 최적제어.



임 충 혁

1987년 서울대 제어계측공학과 졸업. 1989년 동 대학원 석사졸업. 1995년 서울대 제어계측공학과 박사. 현재 서울산업대학교 기계설계자동화 공학부 부교수. 관심분야는 모터제어, 제어공학.



김 동 환

1986년 서울대 기계설계학과 졸업. 1988년 동 대학원 석사졸업. 1995년 Georgia Institute of Technology 박사. 현재 서울산업대학교 기계설계자동화 공학부 교수. 관심분야는 메카트로닉스, 로봇틱스.