

# 비전 기반 청소로봇 시스템에 적합한 로봇의 이동 추적 방법

정문석 윤비에당 이준배 최호철 백성욱  
세종대학교 지능형 미디어 연구실  
msjeong@sju.ac.kr, [n\\_vietthang@sju.ac.kr](mailto:n_vietthang@sju.ac.kr), sbaik@sejong.ac.kr

## A Robust Localization and Orientation Method for Vacuum Robot Generating a Vision System

Moon Seok Jeong, Viet Thang Nguyen, Jun Bae Lee, Ho Chul Choi, Sung Wook Baik  
Intelligent Media Lab, Sejong University

### 요 약

실내의 완벽 주행을 주요 목적인 청소로봇은 자기 위치를 인식 할 수 있어야한다. 자기 위치를 제대로 인식하지 못하게 된다면 청소할 방을 모두 돌지 못하고 방 청소를 마치게 된다. 청소로봇이 상용화 되기 위해서는 저렴한 가격의 보드를 선호한다. 이것은 현재 나온 복잡한 계산을 요구하는 알고리즘을 사용하지 못하거나 사용하여도 속도가 느린 문제를 가진다. 영상 프레임 처리 속도가 느릴 경우 처리되는 동안 로봇이 움직이지 못하여 부드러운 움직임을 불가능하게 한다. 본 논문은 저사양의 하드웨어에서 자기 인식을 할 수 있는 시스템을 제안한다. 자기 인식을 하기 위해 처리 되어야 하는 전처리 과정과 전처리를 거친 데이터를 이용하여 자기 위치를 인식하도록 이동거리와 회전각을 계산하는 방법을 제안한다. 마지막으로 제안된 방법들을 이용하여 실제 이동 값과 비교, 분석한다.

### 1. 서 론

실생활에서 로봇의 의존도가 높아지는 것은 새로운 일이 아니다. 이미 공장에서는 로봇을 이용한 자동화가 이루어졌으며 이제 로봇이 가정으로 들어와 가전기기도 자동화가 이루어지고 있다. 가정에서의 사용이 목적인 로봇으로는 경비로봇, 애완동물 로봇, 청소로봇 등이 있고 이 중 몇몇 제품은 시판되고 있으며 다른 제품들도 곧 시판될 예정이다. 각각의 로봇 연구의 주된 목적으로 경비로봇은 경비해야 할 물체의 상태를 확인하며 물체의 상태가 정상적이지 않을 경우 경고음을 내는 등의 처리를 할 수 있도록 하며 애완로봇의 경우 이동의 제약 해결 보다는 음성인식을 통하여 명령을 받아 해당 명령을 해결 할 수 있도록 하는 것이다. 로봇 제품 중 킬러제품으로 통하는 청소로봇의 경우 장애물에 상관없이 어떤 방이든 방 공간 전체를 완벽하게 이동 할 수 있도록 해야 한다. 이런 로봇들은 더욱 발전하여 사람의 조종 없이 스스로 이동 할 수 있도록 요구되어졌으며 그에 따라 자율주행이 가능한 이동로봇의 개발이 필요하게 되었다.

로봇이 자율 주행을 하는데 있어 중요한 요소로서 로봇은 자기 위치를 추정할 수 있어야 한다.

현재의 자율주행 이동로봇은 초음파, 레이저, 비전을 주로 이용하여 자기 위치를 추정한다. 초음파 센서의 경우 가격이 저렴하고 속도도 빠르지만 정확도가 떨어지며 레이저의 경우 능동적이고 정확하지만 속도가 떨어진다. 비전의 경우 수동적이고 많은 분석이 요구된다.[1] 하지만 비전이 많은 분석이 요구되는 것은 반대로 다른 센서보다 많은 정보를 담고 있다는 뜻이다. 이 정보들은 Harris나 SIFT Algorithm 등의 Detector를 이용하여 정보를 추출하여 장점으로 바뀌었고 그에 따라 자율주행 이동로봇의 연구도 비전을 이용한 것이 많아지고 있다.

비전 시스템을 사용 하는데 있어 일반적인 카메라를 사용하여 비전 시스템을 구축할 경우 좁은 시야각으로 인해 추출할 수 있는 정보가 한정되게 된다. 한정되는 범위는 카메라에 따라 다르지만 대체로 CV-SLAM을 구축하기 위해 천정을 촬영시 천정의 1/4 이상 촬영되지 않는다. 이를 극복하기 위해 어안렌즈를 사용하며 어안렌즈는 다른 대안인 옴니비전 카메라보다 가격이 저렴하

며 부피가 상대적으로 작기 때문에 이동 로봇에 부착도 용이하다. 하지만 어안렌즈는 렌즈의 특수성 때문에 이미지에 왜곡(Distortion)이 심하게 적용되어 정확한 위치 추정이 어렵다. 그러므로 일반 카메라를 사용하는 비전 시스템에서 하지 않는 왜곡을 보정(Calibration)하는 전처리가 추가되어야 한다.

## 2. 관련연구

과거 진행된 비전 연구는 비전 시스템을 이용하여 3차원 공간에서 물체를 인식하여 자기 위치 인식 하는 것에 집중되어 있다. 초기에는 Harris Filter를 이용한 DROID 시스템으로 odometry 정보 없이 로봇 움직임을 알아내었다.[2] 그 후, 다른 시스템과 비전 시스템을 조합하여 맵 빌딩과 자기 위치 인식을 가능하도록 하였다. 그 예로 Expectation Maximization(EM) Algorithm을 적용하여 Solar System과 Vision System을 조합하는 알고리즘으로 맵 빌딩을 하였다. 이 알고리즘을 강화시켜 현재 박물관 안내로봇인 MINERVA가 개발되었으며 Smithsonian's National Museum of American History에서 안내를 하였다.[3] 최근에는 Scale 불변을 이용한 SIFT Algorithm으로 전방에 Bird-Eyed 렌즈를 부착하여 3차원 공간에 있는 물체들을 인지하고 맵 빌딩에 성공하였다.[4][5]

3차원 공간에 대한 연구는 전방의 물체의 변화가 심하기 때문에 천정을 바라보면서 환경의 영향을 적게 받는 연구가 시작되었다. 대표적인 예로 천정과 SIFT Algorithm을 이용하여 맵 빌딩을 하는 CV-SLAM(Ceiling Vision Simultaneous Localization and Mapping)이 있다. [6] 여러 비전 시스템 알고리즘과 비전이 사용하는 카메라의 시선을 바꿔 좀 더 원활하게 자기 위치 인식을 하는 방법들이 존재한다. 하지만 이러한 방법들은 고성능의 PC에서는 원활히 돌아가지만 저성능의 CPU와 카메라를 사용하는 임베디드 환경에서는 속도와 정확도에서 어려움을 가진다. 본 연구는 하드웨어 성능이 낮은 환경에서 자기 위치를 빠르게 하는 방법에 중점을 두었다.

## 3. 특징점에 의존한 자기 위치 추적 방법

로봇이 자기 위치를 추적하기 위해선 우선적으로 특징점의 추출이 전처리 되어야한다. 전처리된 결과는 데이터로 전달되며 자기 위치 추적 시에 사용한다. 본 연구

에서는 이미지를 입력받아 특징점을 추출하여 보정하는 전처리 부분(Preprocessing Part)과 전처리의 결과 데이터를 입력받아 자기 위치를 추적하는 주처리 부분(Main processing Part)으로 구분한다. 전처리 작업에서 이미지를 데이터로 변환한 후 변환된 데이터는 주처리 작업으로 전달된다. 주처리 작업에서는 주어진 연속된 데이터를 이용하여 회전각과 이동거리를 계산한다. 두 결과 값을 비교하여 로봇의 움직임을 결정하고 로봇의 상태를 설정한다. 로봇의 상태와 해당되는 결과 값을 축적하여 로봇의 움직임을 알아낸다.

### 3.1 로봇 움직임 패턴

로봇을 상용화하기 위해서 화질이 떨어지며 가격이 낮은 카메라를 사용한다. 또한, 임베디드 보드의 연산을 담당하는 CPU 성능이 일반 PC보다 떨어지는 제품을 사용한다. 이런 낮은 환경은 연속적인 프레임 처리와 고수준의 알고리즘 사용을 불가능하게 한다. 그러므로 본 연구에서는 로봇의 움직임에 제약을 두며 제약은 다음과 같다. 로봇은 이동과 회전이 동시에 일어나지 않으며 각각 독립적으로 발생한다. 제약에 따라 로봇의 상태를 정의하면 회전(Rotation), 이동(Translation), 정지(Waiting) 그리고 추적 실패(Tracking Error)상태로 구분이 된다.

#### - 정지 (Waiting)

정지 상태는 순차적으로 같은 이미지가 들어와서 계산 결과가 0일 경우를 말한다. 또한 상태가 결정되기 전의 기본 값은 혼란을 막기 위해 정지 상태로 한다.

#### - 회전 (Rotation)

로봇이 멈춘 상태에서 방향을 바꾸게 되면 카메라에서 전송되는 순차적인 이미지들은 y축의 이동이 없으며 이미지 중심으로부터 모델까지의 거리가 동일하며 이미지 중심을 기준으로 회전한 위치에 존재한다.

#### - 이동 (Translation)

이동 상태에서는 로봇은 항상 로봇 위치로부터 전진 혹은 후진만이 가능하다. 이 제한 조건은 카메라에서 전송되는 순차적인 이미지 안의 모델들을 일정 거리만큼 y축을 따라 이동하도록 만든다.

#### - 추적 실패(Tracking Error)

추적 실패 상태는 로봇은 그대로 멈춰있는데 환경적인 요인(태양광, 인공광)으로 인해 순차적인 이미지에 변화가 생겨 이동과 회전이 동시에 일어난 것과 같은 결과가 발생하였을 때 설정된다. 또한 로봇이 계산 할 수 없는 속도로 빠르게 위치 변환을 하였을 경우 역시 추적 실패

로 설정한다.

로봇의 위치는 빠르고 반복적으로 바뀌는 순차적인 이미지를 대상으로 이동과 회전을 계산하고 결과 값을 축적시켜 시작점으로부터 이동 위치를 알아낸다.

그림 1에서 상단은 로봇의 이동을 나타내며 하단은 로봇의 이동에 따른 카메라에 맺히는 이미지의 변화를 나타낸다.

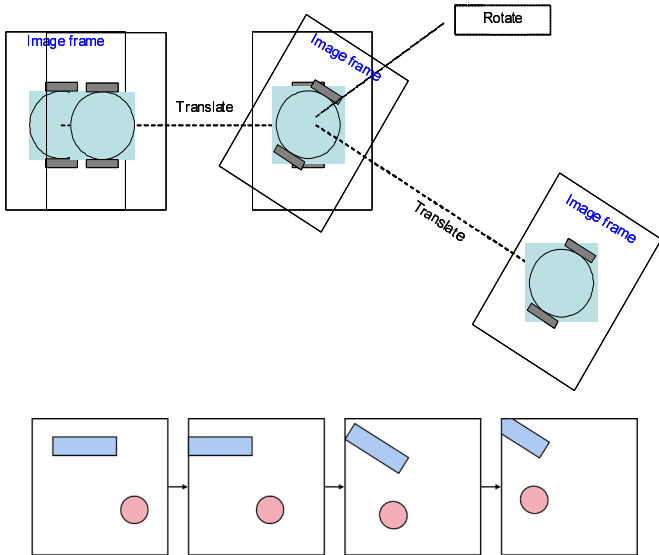


그림 1. 로봇의 이동과 그에 따른 이미지 변화

### 3.2 상태 결정

로봇의 상태는 연속되는 데이터의 회전각과 이동거리를 구한 후 두 상태 사이의 그룹화된 특징점의 매치되는 개수를 비교하여 결정 된다. 로봇의 상태는 데이터가 들어오기 전, 즉 상태가 결정되기 전에는 정지 상태로 설정되며 비교과정을 거친 후 상태를 결정한다. 결정시 회전각과 이동거리의 특징점의 매치되는 개수가 같을 경우 회전상태로 결정한다. 이유는 회전각이 작을 경우 그룹화 과정에서 발생하는 오차로 인해 이동이 이루어졌다고 판단될 수 있기 때문이다. 이동거리가 최소거리에 못 미치고 회전각이 0이면서 양쪽모두 매치되는 특징점이 존재할 경우 정지 상태를 유지한다. 추적 실패의 경우 회전각과 이동거리 모두 threshold 값을 넘어서 매치되는 특징점이 없을 경우 설정한다.

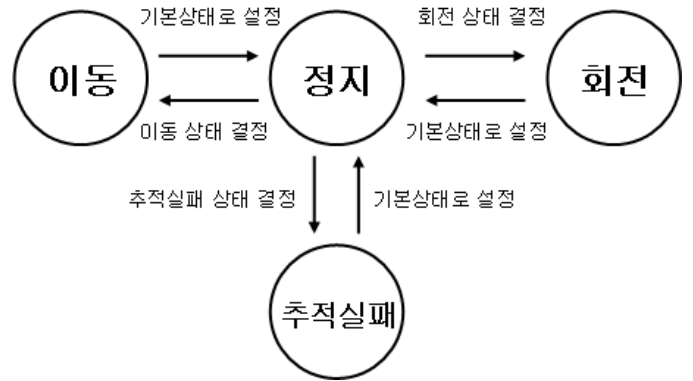


그림 2. State Flow

### 3.3 회전

회전각 계산 방법은 다음과 같은 순서로 이루어진다.

1. 특징점을 추출한다.
2. 특징점이 집중적으로 모여 있는 곳을 그룹화시키고 각 그룹화된 특징점 집합의 중심(Labeling Center Point)을 구한다. 그리고 그림 6과 같이 필요한 정보들 ( $\theta, a$ )을 얻는다.
3. 두 이미지에서 벡터의 길이를 이용하여 같은 그룹을 찾고 각각의 각도 변화를 계산한다. 만약 벡터의 길이가 threshold개수만큼 같은 그룹을 찾지 못하면 로봇은 회전하지 않은 것으로 간주하며 상태를 이동으로 전이시킨다.

4. 모든 그룹간의 각도 변화의 평균을 구하고 그 평균을 로봇의 회전각으로 정한다. 만약 각도가 0도라면 로봇은 회전과 이동을 하지 않은 것으로 간주하며 상태를 정지로 전이시킨다.

그림 3은 여러 개의 그룹 중에서 한 그룹만을 회전각 계산의 예로 보여준다. 상단 이미지는 첫 번째 프레임 특징점 추출 및 벡터 추출이며 하단 이미지는 두 번째 프레임 특징점 추출 및 벡터 추출 이다. 회전 계산 방법에 따르면

- ① 붉은 숫자는 (X축을 기준으로 각도, 원점을 기준으로 거리)를 의미함
- ② 벡터의 크기로 같은 특징점 그룹인지 구분
- ③ 결과(각도의 변화)를 얻음 ( $77 - 51 = 26$ )

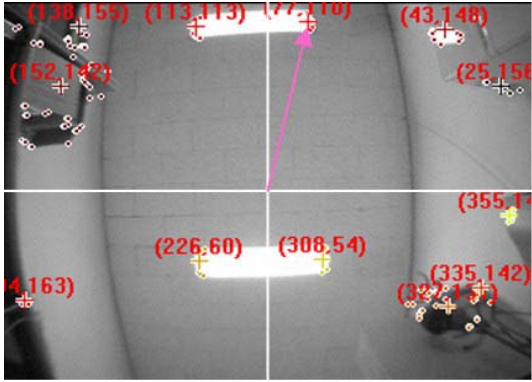


그림 3-1. 회전각 계산. (이동 전)

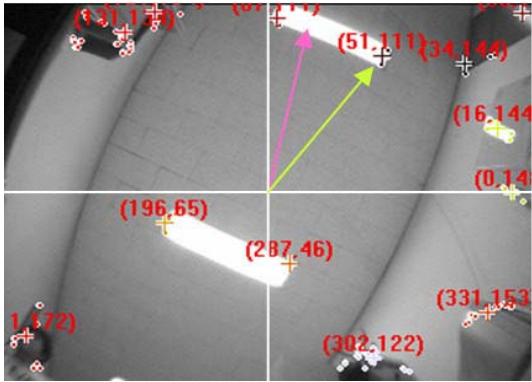


그림 3-2. 회전각 계산. (이동 후)

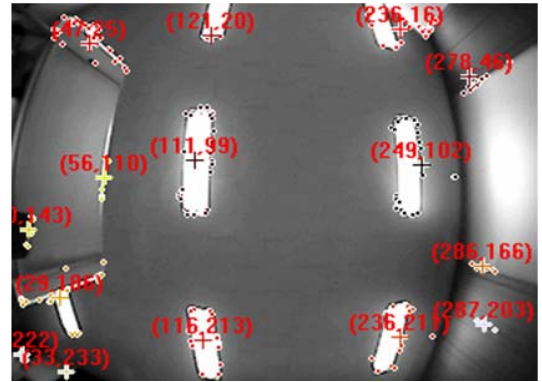


그림 4-1. 이동 거리 계산. (이동 전)

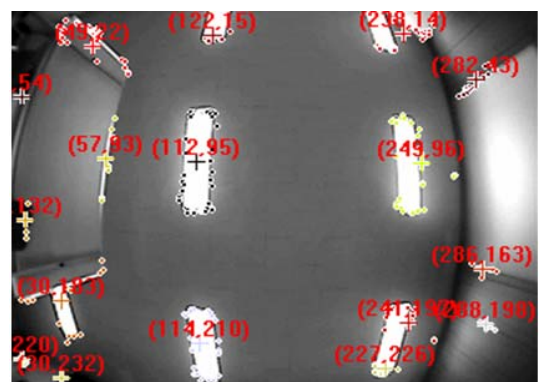


그림 4-2. 이동 거리 계산. (이동 후)

### 3.4 이동

이동 거리 계산 방법은 다음과 같은 순서로 이루어진다.

1. 특징점을 추출한다.
2. 특징점이 집중적으로 모여 있는 곳을 그룹화시키고 각 그룹화된 특징점 집합의 중심(Labeling Center Point)을 구한다. 그리고 그림 7과 같이 필요한 정보들(x,y)을 얻는다.
3. 두 이미지에서 threshold값을 이용하여 같은 그룹을 찾고 각각의 x, y 위치 이동을 계산한다. 만약 x, y의 변화 값이 threshold보다 클 경우 그 그룹은 무시한다.
4. 그룹의 이동 거리를 측정할 수 없다면 상태를 추적 실패 상태로 전이시킨다.

그림 4은 여러 개의 그룹들의 x, y값을 보여준다. 상단 이미지는 첫 번째 이미지 특징점 추출 및 각 그룹의 중심 좌표 추출을 나타내며 하단 이미지는 두 번째 이미지 특징점 추출 및 각 그룹의 중심 좌표 추출을 나타낸다. 또한 붉은 숫자는 (X좌표, Y좌표)를 의미한다.

### 4. 실험 결과

실험은 임의의 실내 거주 공간에서 5회 임의의 경로로 이동하였다. 카메라의 성능은 NTSC 1/3인치 CCD카메라를 사용하였으며 어안렌즈는 대각 150도인 렌즈를 사용하였다. 다양한 실험 결과를 얻기 위하여 실제로 일어나기 힘든 움직임도 포함 시켰다.

방 번호	방 높이(cm)	방 넓이(가로 <sub>(cm)</sub> ×세로 <sub>(cm)</sub> )
1	250	323*424
2	240	280*220
3	195	220*240
4	214	450*675
5	236	1025*895

표 1. 실험 환경 방의 단순 정보표

그림 5은 실험장소인 방 1과 방 2에 관한 평면도이다.

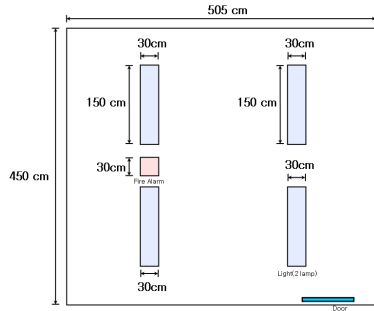


그림 5-1 . 실험 장소 1의 평면도

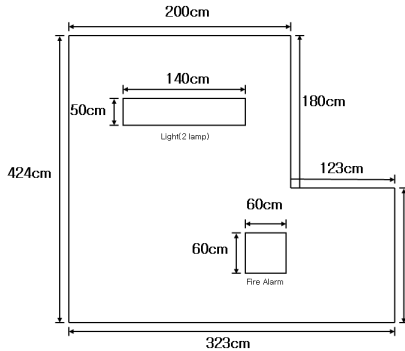


그림 5-2. 실험 장소 2의 평면도

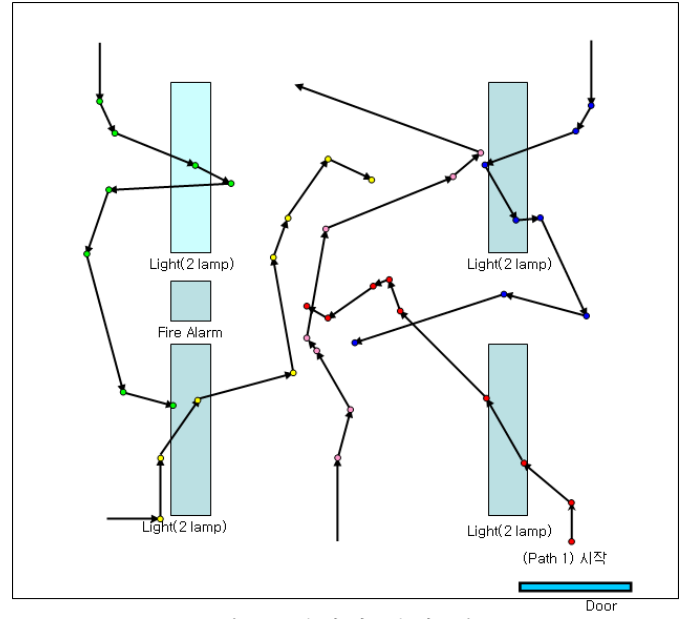


그림 6: 실험한 방의 정보

그림 6은 방 1에서 실험한 다섯 종류의 로봇 이동을 보여준다. 화살표는 진행방향이며 시작과 끝을 제외한 점에서 회전이 발생했다. 표 2는 방 1에서 실험한 결과를 도표로 나타낸 것이다. 결과값은 pixel값으로써 실측과의 비교를 위해서 각 방의 높이상수(1.75)를 입력하여 실제 이동한 거리를 추적하도록 하였다.

실험 횟수	Path 1		Path 2		Path 3		Path 4		Path 5	
	실측	계산값	실측	계산값	실측	계산값	실측	계산값	실측	계산값
1	30cm	31.5 cm	40cm	43.75cm	50cm	52.5cm	45cm	47.5cm	63cm	64.75cm
2	-50°	-47°	-90°	-89°	30°	29°	-25°	-25°	15°	15°
3	50cm	50.75cm	46cm	45.5cm	22cm	22.75cm	27cm	28cm	38cm	38.5cm
4	20°	20°	35°	41°	40°	38°	-43°	-45°	-45°	-46°
5	56cm	54.25cm	50cm	52.5cm	73cm	77cm	66cm	68.25cm	51cm	54.25cm
6	-15°	-13°	40°	40°	-100°	-85°	5°	5°	-10°	-10°
7	92cm	94.5cm	75cm	75.25cm	48cm	49cm	30cm	28cm	10cm	10.5cm
8	25°	24°	-85°	-82°	-65°	-63°	150°	147°	50°	48°
9	24cm	22.75cm	88cm	91cm	18cm	17.5cm	93cm	94.5cm	83cm	87.5cm
10	80°	79°	30°	28°	70°	65°	-68°	-65°	58°	57°
11	14cm	14cm	30cm	31.5cm	82cm	64.75cm	52cm	49cm	105cm	98cm
12	-25°	-24°	15°	15°	130°	91°	-34°	-35°	-18°	-18°
13	46cm	47.25cm	55cm	50.75cm	64cm	64.75cm	109cm	101.5cm	27cm	28cm
14	65°	64°	80°	66°	-33°	-34°	-60°	-59°	-120°	-110°
15	17cm	17.5cm	37cm	40.25cm	120cm	113.75cm	40cm	42cm	150cm	131.25cm
경과시간		15.2sec		15.1sec		15.2sec		15.2sec		15.1sec

표 2. 방 1에서의 로봇 이동 실험 결과

표 3은 선택된 방 다섯 개에서 다른 경로로 실험한 결과를 나타낸다. 최대 오차가 발생하는 이동은 이동거리와 회전각이 클 경우 발생한다. 실제로 로봇이 움직이면서 추적을 한다면 이동 거리와 회전각이 작으면서 빠르게 변하기 때문에 최대 오차까지는 일어나지 않는다.

	이동거리 평균오차	회전각 평균오차	이동거리 최대오차	회전각 최대오차	최대 오차 시 실제 이동	
					이동거리	회전각
Room 1	2.77cm	2.4°	18.75cm	15°	150cm	100°
Room 2	2.84cm	2.4°	19.25cm	14°	125cm	120°
Room 3	2.78cm	2.3°	17.5cm	12°	168cm	110°
Room 4	2.77cm	2.5°	19.25cm	16°	142cm	95°
Room 5	2.91cm	2.6°	21cm	14°	138cm	115°

표 3. 각 방에서의 이동거리 및 회전각 오차

### 5. 결론 및 향후 일정

본 논문에서는 여러 과정을 통하여 자기 위치를 추적하였다. 복잡한 계산은 최대한 줄여서 임베디드 환경에서 실행 될 수 있도록 하였으며 실제로 PXA270보드에서 연속적으로 로봇 위치 변경을 추적하였다. 이러한 자기 위치 추적과 함께 적외선 센서를 함께 사용한다면 적외선 센서의 오차를 비전 센서가 보정할 수 있다고 볼 수 있다.

자기 위치 추적과 더불어 완벽하게 로봇이 자율 주행을 하려면 전방향에 있는 사물을 예측하여야 한다. 카메라가 천정만 바라봐서는 앞에 있는 장애물을 로봇이 예측할 수 없으므로 전방향을 확인 할 수 있는 적외선을 사용하거나 카메라를 설치하여야 한다.

또한 같은 위치에 있다고 하여도 빛의 양에 따라 특징점이 다르게 검출되는 경우가 빈번히 발생한다. 따라서 빛의 양에 무관하게 사물의 특징점을 얻을 수 있는 detector의 개발 혹은 필터를 사용하여 노이즈를 줄이는 방법이 필요하다.

로봇이 천정을 바라볼 수 없는 곳으로 이동할 경우, 예를 들어 탁자 밑으로 들어가서 천정이 가려지게 되었을 때 해결하는 방법도 필요하다.

향후 일정으로는 문제점으로 나타난 천정이 가려졌을 경우 해결하는 방법을 연구하여 특수한 상황에서도 로봇이 이동 할 수 있도록 하며 일반 카메라가 아닌 적외선을 장착한 카메라를 이용하여 주야간의 시스템 성능 차이가 나지 않아야 할 것이다.

### 참고문헌

- [1] 이종실, 민홍기, 홍승홍, “어안렌즈를 이용한 비전 기반의 이동 로봇 위치 추정 및 매핑”, 한국신호처리시스템학회 논문지, 제5권4호, pp. 256-262, 2004년
- [2] C. Harris, “Geometry from visual motion”, Active Vision, pp. 264-284, MIT Press, 1992
- [3] S.Thrun et al., “Minerva: A second-generation museum tour-guide robot”, in Proc. IEEE Int. Conf. Robot. Autom. (ICRA’99), Detroit, MI, May 1999, pp.1999-2005.
- [4] Stephen Se, David G. Lowe and James J. Little. “Vision-Based Global Localization and Mapping for Mobile Robots.” 2005 IEEE Transactions on robotics. Vol. 21 No. 3 June 2005
- [5] S. Thrun, W. Burgard, and D. Fox, “A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping“, IEEE Int. Conf. Robot. Autom.(ICRA’2000) San Francisco, CA, April 2000.
- [6] W. Y. Jeong and K. M. Lee, “CV-SLAM: A new ceiling vision-based SLAM technique”, IROS (International Conference on Intelligent Robots & Systems) 2005, Edmonton, Canada, Aug. 2005