

이동 로봇을 이용한 동시 위치 추정 및 지도 작성에 관한 실험 연구

김혜숙*, 김승연*, 김인철**

*경기대학교 컴퓨터과학과 학부생

**경기대학교 컴퓨터과학과 교수

e-mail:{chia, kimsy0406, kic}@kgu.ac.kr

An Empirical Study on Simultaneous Localization And Mapping with Mobile Robots

Hye-Suk Kim*, Seung-Yeon Kim*, In-Cheol Kim**

*Undergraduate Course, Dept of Computer Science, Kyonggi University

**Faculty, Dept of Computer Science, Kyonggi University

요 약

본 논문에서는 주어진 환경에 대한 정보가 충분하지 않은 상황에서 지능형 에이전트에게 요구되는 스스로의 위치를 파악하기 위해 로봇이 자신의 위치 추정과 동시에 주위 환경을 인식하여 주변 지도를 작성하는 방법을 제안한다. 이동 로봇의 위치를 추정하기 위해 센서 측정값을 통해 계산된 결과 값을 파티클 필터에 적용하며 로봇의 환경 지도 작성을 위해 점유 격자 지도 방법을 사용한다. 이 두 방법을 병합하여 동시적 위치 추정 및 지도 작성 문제에 적용하여 시스템을 설계 및 구현해보고 실험 결과를 소개한다.

1. 서론

지능형 에이전트에게 요구되는 상황 인식(context recognition) 기능 중의 하나가 불확실한 센서 데이터에 의존해 자신의 현재 위치를 파악하는 일이다. 자기 자신의 위치를 파악하기 위해서는 주어진 공간에 대한 정보를 가지고 있는 경우와 그렇지 못한 경우로 나누어 볼 수 있다. 후자의 경우 로봇은 스스로 주어진 공간을 돌아다니며 환경 지도를 작성하는 일을 병행하여 자기 자신의 위치를 파악 하는 수밖에 없다. 이러한 문제를 일반적으로 SLAM(Simultaneous Localization And Mapping)이라 부르며, 환경 지도의 불확실성으로 인해 로봇 측위는 더욱 어려워지게 된다. 본 논문에서는 대표적인 확률 기반의 측위 방법으로 베이지 필터(Bayes filter)의 한 근사방법인 파티클 필터(Particle filter)를 이용해 로봇이 자기 자신의 위치를 추정하는 방법을 적용하여 보았고, 환경에 대한 지도를 작성(mapping)하기 위해서 센서의 불확실성을 고려하여 효과적으로 새로운 공간을 표현하기 위해 자유 공간과 물체를 함께 표시 할 수 있는 복합 공간 지도(Composite Space Map)의 한 종류인 점유 격자 지도(Occupancy Grid Map) 표현법을 이용하여 보았다. 위의 두 방법을 사용하여 구현된 동시적 위치 추정 및 지도 작성법을 소개한다.

2. 동시적 위치 추정 및 지도 작성

로봇 측위(Robot Localization)는 측정된 센서 데이터와 환경 지도상의 물체들의 절대 위치를 이용해 환경 지도상에 로봇이 현재 어디쯤 위치하고 있는지를 추정하는 문제와 로봇이 충분히 경험하지 못한 새로운 공간인 경우로 나뉜다. 환경 지도를 가지고 있지 못한 경우, 로봇 스스로 주어진 공간을 돌아다니며 환경 지도를 작성하는 일(Mapping)과 로봇 측위를 병행하는 문제를 일반적으로 SLAM이라 부른다.

로봇측위의 대표적인 확률기반의 측위 방법으로 베이지 필터(Bayes filter)가 있다. 베이지 필터는 로봇 센서와 이동모터의 불확실성(uncertainty)을 고려하여 로봇의 위치에 대한 사후 확률 분포(posterior probability distribution)을 계산한다. 파티클 필터는 t 시간에 물체의 위치 x_t 에 대한 사후 확률 분포 $bel(x_t)$ 를 유한개의 샘플링 된 파티클들로 표현하며, 이 파티클들은 각각 t 시간에 물체가 존재할 가능성이 큰 후보 위치를 나타낸다.

복합지도의 한 종류인 점유 격자 지도표현법은 전체공간을 일정한 크기의 작은 셀(cell)들로 나누고, 각 셀별로 그 셀 영역 위에 장애물이나 물체가 점유하고 있을 가능성을 0과 1의 이진값(binary value)이나 확률값(probability)등으로 나타낸다.

<표 1>의 알고리즘은 크게 3단계로 구성된 과정을 반복한다. 예측 단계(Prediction step)에서는 이동 제어 u_t 로 인해 이전 위치 x_{t-1} 에서 새로운 위치 x_t 로 이동할 가능

※ 본 연구는 경기도의 경기도지역협력연구센터사업의 일환으로 수행하였음

성을 나타내는 이동 모델(motion model) $p(x_t|x_{t-1}^{[m]}, u_t)$ 에 따라 파티클 집합에 속한 파티클들을 이동시켜 새로운 파티클들을 구한다. 갱신 단계(Update step)에서는 30도씩 12번에 걸쳐 총 360도를 회전하면서 주변 장애물까지 거리를 측정한다. 그리고 센서가 한번 스캔하는 범위 안의 셀에 물체가 있으면, 해당 셀의 점유 도를 1 증가시키고, 물체가 없으면 1 감소시킨다. 점유도가 높은 셀 일수록, 해당 셀에 장애물이나 벽이 존재할 가능성이 더 높다고 판단하며 이러한 과정을 반복하며 맵이 갱신되어진다. 12번의 센서측정치 중 첫 번째 측정치 z_t^1 와 현재 위치 x_t 에서 센서 측정치 z_t 를 얻을 가능성을 나타내는 인식 모델(Perceptual model) $p(z_t^1|x_t^{[m]})$ 을 이용하여 새로운 위치 상태에 대한 가중치(weight) $w_t^{[m]}$ 를 계산한다. 리샘플 단계(Resample step)에서는 가중치 $w_t^{[m]}$ 에 비례하는 확률에 따라 새로운 파티클 집합에 포함될 파티클 $x_t^{[i]}$ 을 선정한다.

<표 1> SLAM의 기본 알고리즘

```

Algorithm SLAM( $X_{t-1}, u_t, z_t, md_{t-1}, v_{t-1}$ )
 $\hat{X}_t = X_t = \{\}$ , M=Number of particles
for m=1 to M do
  // Prediction Step
  Sample  $x_t^{[m]}$  with probability  $p(x_t|x_{t-1}^{[m]}, u_t)$ 
  // Update Step
  plotSquares( $X_t, z_t, md_t$ )
  //update mapdata
endfor
 $w_t^{[m]} = p(z_t^1|x_t^{[m]})$ 
 $\hat{X}_t = \hat{X}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle$ 
endfor
for m=1 to M do
  // Resample Step
  Draw i with probability  $\propto w_t^{[m]}$ 
  Add  $x_t^{[i]}$  to  $X_t$ 
endfor
return  $X_t$ 
    
```

3. 실험 환경 구성

본 논문에서는 실제 이동 로봇을 이용하여 앞서 설명한 동시적 위치 추정 및 지도 작성에 대한 시스템을 설계하고, 세 개의 경로를 지정하여 각 경로별로 실험을 전개하였다. 실험에 사용된 로봇의 구성은 (그림 1)과 같은 Lego Mindstorm NXT이며, 로봇의 제어 프로그램은 JAVA API를 기초로 하는 leJOS NXT를 이용하여 구현하였다.



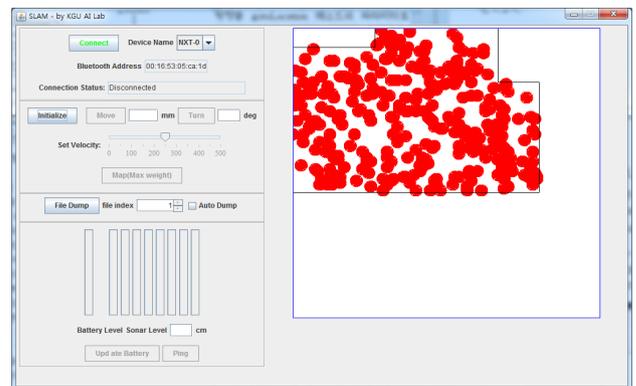
(그림 1) 실험용 로봇

(그림 1)의 로봇에는 총 세 개의 모터와 한 개의 센서를 사용하였다. 센서는 바로 앞의 장애물까지의 거리를 측정할 수 있는 초음파 센서(Ultrasonic Sensor)를 사용하였고, NXT brick 위에 있는 1개의 모터는 센서를 12회 돌려 전방향 탐색(omnidirectional exploration)을 하는데 사용하였고, 나머지 2개의 모터는 로봇의 이동(move) 동작과 회전(turn) 동작을 위하여 사용하였다.



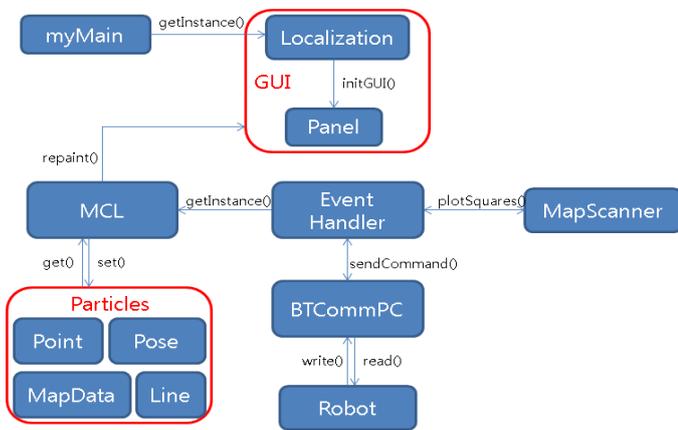
(그림 2) 추정 및 작성 환경

로봇의 위치 추정 및 지도 작성을 위한 실험 환경은 (그림 2)과 같이 구성하였으며 (그림 1)의 이동 로봇이 사용자의 명령에 따라 환경 안에서 움직이며 위치 추정 및 지도를 제작한다.



(그림 3) 프로그램 실행 화면

동시적 위치 추정 및 지도 작성을 위한 전체 시스템은 크게 이동 로봇과 PC 서버로 구성되며, 로봇과 PC 서버는 블루투스(Bluetooth)를 이용하여 무선으로 연결되어 있다. 사용자는 PC 서버에 해당하는 (그림 3)의 사용자 인터페이스(User Interface)를 통해 이동 로봇의 움직임을 제어하며 실제 지도의 모양 및 움직임에 따른 센서의 측정값, 파티클의 방향, 로봇의 위치 추정 과정 그리고 현재 시간에서 가중치가 가장 높은 파티클이 작성중인 지도를 확인할 수 있다. (그림 4)는 PC 내부에서 실행되는 프로그램의 구성도이다. 사용자가 실험 공간에 이동 로봇을 위치시킨 후 사용자 인터페이스에서 로봇의 이름을 선택하면 해당하는 로봇의 맥 주소(Mac address)가 나타나게 된다. 사용할 로봇과 맥 주소를 확인 후 Connect 버튼을 누르면 (그림 4)의 BTCommPC 클래스에서 이동 로봇과 연결을 시도하고 로봇과 PC 서버간의 연결이 완료되면 Connect 버튼이 Disconnect 버튼으로 변화한다.



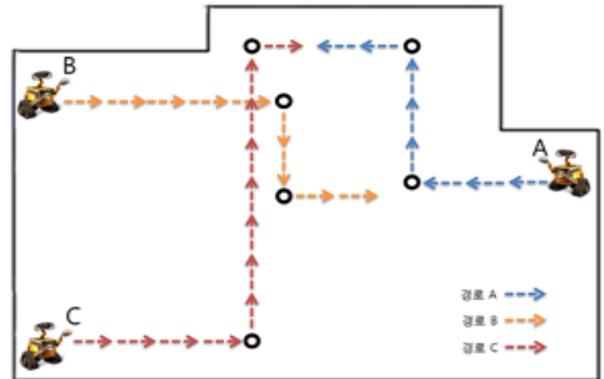
(그림 4) 프로그램 구성도

사용자가 Move 혹은 Turn 버튼을 누르면 사용자가 내린 명령이 EventHandler 클래스를 통해 BTCommPC 클래스에 전달되고 최종적으로 로봇이 명령을 수행하게 된다. 사용자의 이동 혹은 회전 명령이 수행되고 나면 이동 로봇 내에 내장된 Robot 클래스안의 ping() 메소드가 실행되어 전방의 장애물까지의 거리를 측정한다. 측정값이 얻어지면 scanCircle() 메소드가 수행된다. 이 메소드는 전 방향의 거리를 측정하며 scanCircle() 메소드의 실행이 종료되면 명령의 결과값에 해당하는 센서 측정값을 writeInt() 메소드와 writeDouble() 메소드를 이용하여 BTCommPC 클래스로 보낸다. 센서 측정값은 BTCommPC 클래스를 통해 EventHandler 클래스에서 사용된다. 센서 측정 값 중 첫 번째 값은 MCL 클래스에서 로봇의 위치 추정을 위해 사용되며 나머지 12개의 센서 측정값은 추정 중인 위치 값과 이전의 맵 데이터와 함께 MapScanner 클래스의 plotSquares() 메소드에 매개변수로 사용되어 지도를 작성하는 가중치의 값을 계산하여 최종적으로 파티클 별 지도를 작성하도록 한다. 그리고 MCL 클래스에서 현재까

지의 가중치를 바탕으로 파티클을 리샘플링 한다. 모든 과정을 종료 후 Particles 클래스 내의 모든 파티클의 위치 정보, 가중치 및 지도 정보가 갱신되며 가중치가 가장 높은 파티클이 작성중인 지도는 사용자가 Map 버튼을 클릭 시 사용자 인터페이스를 통해 PC 화면에 나타난다.

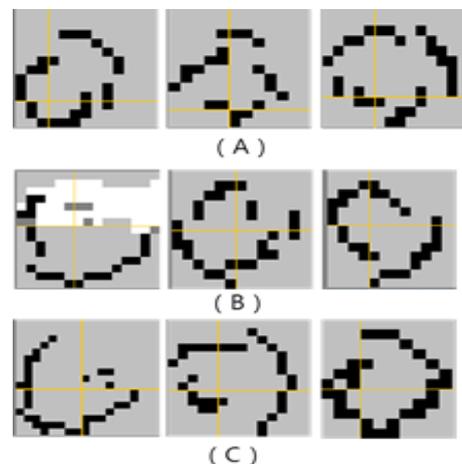
4. 구현 및 실험

앞서 설명한 동시적 위치 추정 및 지도 작성 시스템을 실험하기 위해 같은 시스템을 각기 다른 세 가지 이동 경로를 이용하여 수행하였다.



(그림 5) 로봇의 이동 경로

(그림 5)는 사용자가 로봇을 제어하는 이동 경로 세 가지를 표현하고 있다. 파랑색 화살표는 경로 A, 노란색 화살표는 경로 B, 빨강색 화살표는 경로 C를 나타내며 각 화살표는 이동 로봇이 바라보는 방향으로 이동 동작을 수행하였음을 의미하고 화살표 사이의 점은 회전 동작을 수행하였음을 의미한다. (그림 6)은 (그림 5)의 세 가지 경로를 사용자의 명령에 따라 로봇이 움직이며 작성한 지도 중 가중치가 가장 높은 파티클의 중간1, 중간2, 결과에 해당하는 데이터이다.



(그림 6) 경로별 생성된 격자 지도

지도 화면에 흰색으로 표시된 부분은 가중치가 매우 낮은 셀이며 이는 장애물이 없는 셀로 볼 수 있다. 검은색으로 표시된 부분은 가중치가 매우 높은 셀로 장애물이 있다고 판단한 셀이다. 지도 내에 보이는 노란 선이 교차된 지점은 지도 안에서 자신의 위치를 나타낸 좌표를 나타낸다. 경로 A, B, C에서 중간1의 지도에 비해 결과 지도가 더 자세한 이유는 파티클의 가중치가 높은 즉 정확한 위치 추정이 이루어 졌기 때문에 지도의 정확성이 높아졌다고 볼 수 있다.

5. 결론

본 논문에서는 불확실한 센서 데이터와 공간에 대한 부족한 정보를 극복하기 위하여 동시적 위치 추정 및 지도 작성 방법을 제안하였고 이를 설계 및 구현하였다. 이를 Lego Mindstorm NXT로 제작된 로봇을 사용하여 로봇의 위치 추정 및 지도 작성 실험을 하여 방법의 성능 확인을 위한 실험의 결과를 소개하였다. 향후 현재 제안한 시스템의 성능을 개선하기 위해 현재 파티클의 가중치를 정하는 방법은 Localization에만 의존적인 부분에서 탈피하여 지도 작성과 함께 작성된 지도 내에서 계산된 로봇의 위치가 얼마나 적합한지에 대하여 가중치를 주는 새로운 전략을 개발하는 방향으로 연구를 진행할 계획이다.

참고문헌

- [1] Hugh Durrant-Whyte and Tim Bailey, "Simultaneous Localization and Mapping (SLAM): Part I The Essential Algorithms", Robotics and Automation Magazine, Vol. 13, 2006.
- [2] Jeong Woo, et al, "Localization of Mobile Robot using Particle Filter", Proc. of SICE-ICASE-2006, 2006.
- [3] Brian Bagnall, Intelligence Unleashed: Creating LEGO NXT Robots with Java: CREATING LEGO NXT ROBOT WITH JAVA, Variant Press, 2011.
- [4] Brian Bagnall, Maximum Lego NXT: Building Robot with Java Brains, Variant Press, 2007.