

〈 논문 〉

바코드가 있는 가정환경에서의 위상학적 지도형성 및 자율주행

허진욱* · 정웅식** · 정완균†

(2006년 3월 27일 접수, 2006년 6월 23일 심사완료)

Topological Mapping and Navigation in Indoor Environment with Invisible Barcode

Jinwook Huh, Woong Sik Chung and Wan Kyun Chung

Key Words: Invisible Barcode(바코드), Mobile Robot(이동로봇), Navigation(자율주행), Topological Mapping(위상학적 지도)

Abstract

This paper addresses the localization and navigation problem using invisible two dimensional barcodes on the floor. Compared with other methods using natural/artificial landmark, the proposed localization method has great advantages in cost and appearance, since the location of the robot is perfectly known using the barcode information after the mapping is finished. We also propose a navigation algorithm which uses the topological structure. For the topological information, we define nodes and edges which are suitable for indoor navigation, especially for large area having multiple rooms, many walls and many static obstacles. The proposed algorithm also has an advantage that errors occurred in each node are mutually independent and can be compensated exactly after some navigation using barcode. Simulation and experimental results, were performed to verify the algorithm in the barcode environment, and the result showed an excellent performance. After mapping, it is also possible to solve the kidnapped case and generate paths using topological information.

1. 서 론

최근 이동로봇(Mobile Robot) 기술은 청소용 로봇, 교육용 로봇 등 가정에서 볼 수 있는 서비스 로봇부터 군사용 로봇에 이르기까지 활용범위가 점차 범위가 넓어지고 있다.^(1,2) 이동로봇 연구 분야에서 위치인식(Localization)과 함께 중요한 부분을 차지하고 있는 것이 자율주행(Navigation)과 지도형성(Mapping) 분야이다. 이중 자율주행과 지도형성은 서로 떨어뜨려 생각하기 어려운 정도로

밀접한 관계가 있다. 자율주행을 위해서는 지도가 필요하고 지도형성을 위해서는 로봇이 스스로 지도형성을 할 수 자율주행 능력이 필요하기 때문이다. 그렇지만 현재의 자율주행 기술은 로봇이 스스로 지도를 형성해서 이동하기보다는 미리 프로그램화된 기능만을 수행하거나 장애물을 회피하는 정도의 수준밖에 되지 않는다. 특히 저가의 로봇들이 지도를 형성하지 못하고 정형화된 움직임 밖에 할 수 없는 이유는 크게 두 가지가 있다. 첫째로 로봇 스스로 자신의 위치를 인식하기 위해서는 복잡한 위치인식 알고리즘이 필요하다.^(3,4) 그렇지만 현재의 상업용 로봇에는 이 알고리즘을 적용하는데 어려움이 있다.

두 번째로 가정환경은 벽이나 장애물들로 인해 복잡한 구조를 가지고 있어서 로봇이 스스로 경로를 형성하며 이동하기에는 많은 문제가 있다.

† 책임저자, 회원, 포항공과대학교 기계공학과 교수
E-mail : wkchung@postech.ac.kr
TEL : (054)279-2172 FAX : (054)279-8459

* 국방과학연구소(jwhuh@postech.ac.kr)

** 마이크로로봇(newkid72@microrobot.co.kr)

이러한 문제를 해결하고 로봇이 스스로 가정환경의 지도 및 경로를 형성하는 자율주행의 능력을 가진다면 현재의 단순한 기능을 넘어서 상위의 지능을 필요로 하는 기능들을 하게 된다. 그러므로 로봇이 가정환경의 다양한 변화에 상관없이 스스로 정확한 지도를 형성하고 그 지도를 바탕으로 자율주행을 할 수 있는 알고리즘 개발은 이동로봇 분야에서 매우 중요하다. 가정환경에서 로봇이 스스로 지도형성을 하는 알고리즘들은 많은 논문에서 제안되었다.^(5,6) 그렇지만 이 알고리즘들은 복잡한 가정환경에 적용하기에는 너무 비효율적이다.

본 연구에서는 앞에서 설명한 두 문제를 기존의 알고리즘들과 달리 다음과 같이 해결하였다. 먼저 위치인식 문제는 아파트 바닥에 눈에 보이지 않는 바코드가 인쇄된 바닥재를 사용해서 해결하였다. 그리고 새로운 노드(Node)와 에지(Edge)의 정의를 통해 가정환경에 적합한 위상학적 구조(Topological Structure)를 가짐으로써 다양하고 복잡한 가정환경에서의 경로형성 및 자율주행이 가능하도록 하였다. 이 방법들은 가정환경에서 위상학적 구조를 이용한 기존 알고리즘⁽⁷⁻⁹⁾들과 달리 저가의 센서를 이용할 수 있다는 특징이 있다.

복잡한 가정환경에서 초기 지도형성부터 주행까지 모든 과정을 자율적으로 행할 수 있는 알고리즘을 본 논문에서 제시하고자 한다.

2. 위치인식 기법

한국이나 일본 또는 몇몇 유럽 국가에서는 대부분의 가정이나 사무실에 바닥재를 사용하고 있다. 이런 환경에 적합하게 위치인식을 하고자 눈에 보이지 않는 바코드를 인쇄된 특수 바닥재를 개발하였다. 바코드를 이용해 위치인식을 하는 것은 기존의 인공적 표식을 이용하는 것과는 다르다.^(10,11) 기존의 방법들은 벽이나 천장에 인식할 수 있는 표식을 붙이기 때문에 미관상 좋지 않고 추가적 비용이 들지만 바코드 표식은 이러한 문제들을 해결하였다. 이 장에서는 바코드 표식을 이용한 위치인식 기법에 대해 설명하고자 한다.

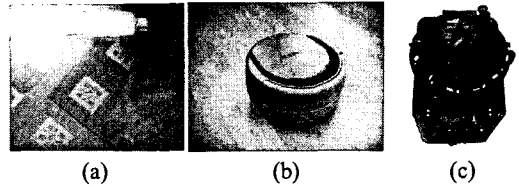


Fig. 1 System description

2.1 시스템 구성

2.1.1 바닥재의 특징

바닥재는 다음과 같은 특징을 가지고 있다.

(1) Fig. 1(a)와 같이 육안으로 봤을 때 일반적인 바닥재와 차이가 없으나 자외선을 쬐었을 때 바코드가 나타난다.

(2) 바닥재 각 조각들은 7.5cm×90cm의 크기이고 각 조각들은 2열 24행의 바코드 행렬을 가지고 있다.

(3) 각 바코드는 X, Y 값을 가지고 있고 첫 번째 열에서의 X의 값은 임의의 짝수 값이고 두 번째 열의 X의 값은 첫 번째 열의 X의 값에 1을 더한 홀수 값이다. 그리고 Y의 값은 같은 조각 내에서 0에서 23까지 단순증가 한다. 바코드와 로봇의 상대적 각도와 거리를 알 수 있다.

(4) 각 바닥재 조각들 사이에는 Y방향으로 3개 바코드 차이로 붙인다는 것 외에는 다른 규칙은 없다.

위와 같은 바닥재는 몇 장점이 있다. 먼저 바코드를 인쇄하는데 잉크 값을 제외하고 비용이 거의 들지 않고 또한 바코드가 육안으로는 보이지 않으므로 다른 인공적 표식의 외관상 좋지 않다는 단점을 극복하였다. 그렇지만 바닥재 조각들은 시공자에 의해 임의로 배열되므로 절대 위치를 가르쳐주지 않는다. 즉 바코드는 위치인식을 위한 표식이 될 뿐이므로 바코드 정보를 이용해서 위치인식을 할 수 있는 기법이 필요하다.

2.1.2 실험용 로봇

실험에 사용된 로봇은 Fig. 1(b), Fig. 1(c) 두 종류가 사용되었다. Fig. 1(b)는 청소기 로봇 개발을 위한 실험용 로봇으로 IR센서 종류인 Sharp사의 GP2D120(인식거리 4~30cm) 7개를 사용하였고 Fig. 1(c)는 BSRI(Brilliant Service Robot I)로 차륜구동형 이동로봇이고 실험에 사용된 센서는 GP2D120 다섯 개와 GP2Y0AO2YK(인식거리 20~150cm) 두 개를 사용했다. IR센서 두 모델 모두 \$10 미만의 저가 센서이고 전방에 30도 간격으로 배치하였다.

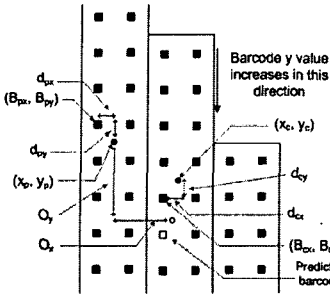


Fig. 2 Correction of the robot position

2.2 위치인식 기법

본 연구에서는 위치인식을 위하여 바코드와 오도메트리(Odometry) 정보를 이용하였다. 위치인식은 예측단계(Prediction Step)와 보정단계(Correction Step)로 나뉘지고 과정은 Fig. 2와 같다. 그리고 위치인식에 사용되는 기본 변수들은 다음과 같다.

- (Ox, Oy) : 오도메트리 정보
- (Bpx, Bpy): 이전 바코드의 x, y 값
- (Bcx, Bcy): 현재 바코드의 x, y 값
- DGx, DGy: 바코드 사이 간격. 여기서는 3.75cm로 일정
- Npx, Npy : 로봇이 x, y방향으로 지나쳤다고 예측되는 바코드의 개수
- (dpx, dpy) : 이전에 읽은 바코드와 이전 로봇위치와의 거리
- (dcx, dcy) : 현재 읽은 바코드와 현재 로봇위치와의 거리
- Lx, Ly : 이전에 읽은 바코드와 예측된 바코드 사이의 거리
- Mpx, Mpy : 예측된 현재의 바코드 모드
- Mcx, Mcy : 현재 읽은 바코드 모드
- (xp, yp) : 이전 로봇 위치
- (xc, yc) : 보정된 현재 로봇 위치

여기서 N_{px} 와 N_{py}는 정수 값이다. 그리고 [x]는 x보다 크지 않은 정수이다. $a \equiv b \pmod m$ 는 두 정수 a와 b는 modulo m에 대해 합동이라는 뜻으로 a와 b는 m으로 나눴을 때 같은 나머지를 갖는다. (d_{px}, d_{py})와 (d_{cx}, d_{cy})는 바코드 리더기에서 주는 값이다. 먼저 N_{px}와 N_{py}를 오도메트리 정보만을 이용해서 계산한다.

$$N_{px} = \left\lfloor \frac{O_x + d_{px} - d_{cx}}{D_{Gx}} \right\rfloor, N_{py} = \left\lfloor \frac{O_y + d_{py} - d_{cy}}{D_{Gy}} \right\rfloor$$

바닥재들은 Fig. 2 와 같이 y방향으로 3개의 바코드 차이로 배열되기 때문에 바닥재 경계를 넘을 때마다 바코드 y의 값은 3씩 증가하거나 감소하게 된다. 그러므로 M_{px}는 다음과 같이 된다.

$$M_{px} \equiv (B_{px} + N_{px}) \pmod 2 \quad 0 \leq M_{px} < 2$$

M_{py}는 두 가지 경우로 나누게 된다. 한 바닥재

Table 1 α, β value for updating robot position

$M_{px} - M_{cx}$	$M_{py} - M_{cy}$	α	β
0	0	0	0
	1	0	D_y
	-1	0	$-D_y$
$(M_{px} = 0)$	0	D_x	0
	1	D_x	D_y
	-1	D_x	$-D_y$
	-3	$-D_x$	0
	-4	$-D_x$	$-D_y$
	-2	$-D_x$	D_y
1	3	D_x	0
	4	D_x	D_y
	2	D_x	$-D_y$
	-1	$-D_x$	$-D_y$
	0	$-D_x$	0
	1	$-D_x$	D_y

에는 바코드가 2열로 배열되어 있고 경계를 넘어갈 때마다 y의 값이 3이 증가하거나 감소하므로 이전에 읽은 바코드 x mod 값이 짝수인지 홀수인지에 따라 y mod 값이 다음과 같이 된다.

$B_{px} \equiv 1 \pmod 2$ 인 경우

$$M_{py} \equiv \left(B_{py} + N_{py} + 3 \times \left\lfloor \frac{N_{px} + 1}{2} \right\rfloor \right) \pmod{24}$$

$$0 \leq M_{py} < 24$$

$B_{px} \equiv 0 \pmod 2$ 인 경우

$$M_{py} \equiv \left(B_{py} + N_{py} + 3 \times \left\lfloor \frac{N_{px}}{2} \right\rfloor \right) \pmod{24}$$

$$0 \leq M_{py} < 24$$

다음 단계는 예측되는 현재의 바코드 값과 현재 읽은 바코드 값을 이용해서 현재의 로봇의 위치를 보정하는 보정단계(Correction Step)이다. 먼저 이전 바코드와 d_{px}, d_{py}와 오도메트리 정보를 가지고 현재 예측된 바코드와 이전 바코드 사이 거리 L_x, L_y를 구하는데 식은 다음과 같다.

$$L_x = \left[\frac{O_x + d_{px} - d_{cx}}{D_{Gx}} \right] \times D_{Gx}$$

$$L_y = \left[\frac{O_y + d_{py} - d_{cy}}{D_{Gy}} \right] \times D_{Gy}$$

바코드 리더기는 현재 바코드 정보 x, y와 현재 바코드와 로봇과의 거리인 d_{cx}, d_{cy} 값을 주는데 이 값들을 이용해서 현재의 바코드의 mod 값을 다음과 같이 얻을 수 있다.

$$M_{cx} \equiv B_{cx} \pmod{2} \quad 0 \leq M_{cx} < 2$$

$$M_{cy} \equiv B_{cy} \pmod{24} \quad 0 \leq M_{cy} < 24$$

그리고 현재의 보정된 로봇 위치는 다음과 같이 계산이 된다.

$$x_c = x_p + L_x + (d_{cx} - d_{px}) + \alpha$$

$$y_c = y_p + L_y + (d_{cy} - d_{py}) + \beta$$

윗 식에 있는 α 와 β 는 예측된 바코드와 실제 바코드와의 차이를 보정하는 값으로 Table 1에 나와 있다. Fig. 2에서 보듯이 오도메트리 값은 조금씩 오차가 존재하므로 예측된 바코드의 위치는 실제 바코드의 위치에서 조금씩 벗어날 수 있다. 그래서 예측된 바코드는 실제로 읽힌 바코드의 값과 조금씩 차이가 나게 되는데 이것을 α 와 β 로 보정하는 것이다. 이렇게 매 순간마다 바코드가 로봇의 위치를 가르쳐주지는 않지만 위와 같은 방법으로 현재의 오도메트리 정보를 바코드로 보정함으로써 로봇의 정확한 위치를 알 수 있다.

3. 효율적인 지도형성 방법

일반적인 가정환경은 커다란 거실과 몇 개의 방으로 구성 되어 있고 각 방과 거실들은 벽이나 고정된 장애물들로 나누어져 있다. 그리고 각 영역들은 다른 영역으로 넘어갈 수 있는 특징 있는 영역을 가지고 있으며, 이것은 문이 될 수도 있고 고정된 장애물에 의해 생긴 영역일 수도 있다. 그러므로 이러한 가정환경에서의 특징들을 이용하여 각 공간 사이의 상대적 정보를 갖는 위상학적 지도(Topological Map)를 구성할 수 있다.

3.1 가정환경에서의 Node와 Edge 정의

먼저 지도형성 및 자율주행을 위해 필요한 가정을 다음과 같이 하고자 한다.

- 가정 1. 가정환경에서 방과 방 사이를 넘나들 수 있다
- 가정 2. 로봇은 위치인식이 가능하다
- 가정 3. 지도형성 해야 할 공간의 크기는 정해져 있다
- 가정 4. 지도형성 중에는 장애물을 이동하는 경우는 없다
- 가정 5. 지도형성시 로봇이 장애물에 막혀 다른 영역으로 이동 못하는 경우는 없다

일반적인 위상학적 구조에서 에지는 특징이 있는 위치인 노드를 연결하는 것에 불과하지만 이

논문에서 제시하는 위상학적 구조에서는 에지가 노드보다 더 중요한 의미를 가지므로 먼저 에지에 대해 정의하고자 한다. 에지에 대한 정의는 다음과 같다.

정의 1. Edge

한 영역과 다른 영역을 연결하는 위치로 그 폭은 Robot이 지나갈 수 있으나 제한된 폭을 가지고 있는 위치를 Edge라고 한다. 다음을 만족하는 위치가 Edge가 된다.

- $\frac{3}{2}R < W < L$ (R : 로봇반지름)
- 노드와 노드를 연결한다.

대부분의 나라에서 문 폭은 0.7m에서 1.5m에 해당되므로 본 연구에서는 L 의 값으로 1.5m을 선택하였다. 이와 같이 에지를 정의했을 때 에지들은 다음과 같은 몇 가지 특징을 가지고 있다.

성질 1. Edge

- Edge의 양쪽 끝은 각각 다른 노드(Node)로 되어 있다.
- 양쪽 끝에 연결된 노드의 index가 같고 그 위치가 같으면 같은 Edge이다.
- 한 영역을 연결하는 edge는 여러 개 존재할 수 있다.

위와 같은 성질은 공간상 에지의 양쪽 끝이 같은 영역일 수는 없기 때문에 나타나는 것이다. 그러므로 에지의 양쪽 끝은 각각 다른 노드로 연결이 되어야 한다. 이와 같은 에지의 성질 때문에 뒤에서 설명할 데드엔드 에지(Dead-End Edge), 퓨전 에지(Fusion Edge)가 제거 된다. 노드에 대한 정의는 다음과 같다.

정의 2. Node

벽이나 고정된 장애물로 둘러싸인 영역으로 Edge를 통해서 다른 Node로 이동할 수 있는 공간을 Node라고 한다.

노드가 가지고 있는 정보는 노드 번호(Node Index), 연결된 에지 정보(Data of linked edge), Map 정보(Local Grid Map)를 가지고 있다. 위와 같이 노드를 정의했을 때 노드는 다음과 같은 성질을 갖는다.

성질 2. Node

- 각 영역들은 독립된 공간들을 가지고 있다

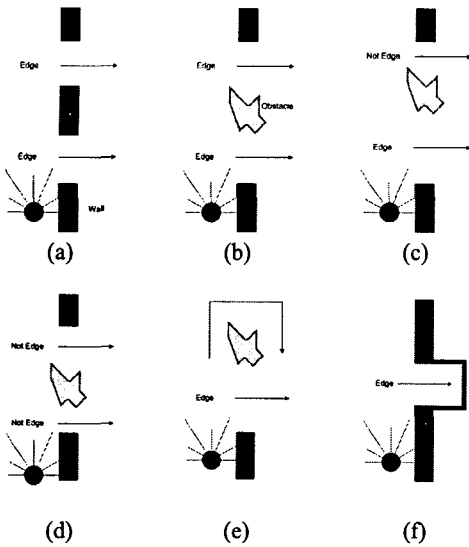


Fig. 3 Various cases defined edge

3.2 Edge의 생성, 제거 및 병합

앞에서 설명한 것처럼 새로운 노드와 에지 대한 정의를 통해 가정환경에서 위상학적 구조를 구성할 수 있다. 그렇지만 에지로 정의할 필요가 없는 부분에서 많은 부분들이 에지로 정의되는 경우가 많다(Fig. 3 참조).

Fig. 3(a)는 다른 노드로 넘어가는 에지가 두개가 존재하는 경우이다. 예를 들어 방과 방 사이에 문이 두 개 있는 경우이다. 이 경우에는 로봇이 두 개의 에지를 생성하게 된다.

Fig. 3(b)는 원래는 에지가 하나이지만 에지 사이에 장애물이 놓여 있어서 로봇이 두 개의 에지가 있다고 생각이 되는 경우이다. 로봇은 두 개 중 어느 것을 이용하더라도 다른 에지로 이동이 가능하기 때문에 문제가 되지 않는다.

Fig. 3(c)는 Fig. 3(b)와 같이 에지 사이에 장애물이 놓여 있어서 기존의 에지의 위치가 약간의 변동했다는 것 외에는 변화가 없다.

Fig. 3(d)는 기존에 에지가 한 개 있었는데 에지 사이에 장애물이 놓여서 로봇이 에지를 인식하지 못하는 경우이다. 즉 이 에지를 통해 로봇이 다른 노드로 이동하는 것은 매우 위험한 것이고 처음에 가정했던 것에 어긋난다(가정 5). 이러한 경우는 지도형성 중에 존재해서는 안 된다.

Fig. 3(e)는 에지가 존재하지 않는 경우이다. 그러나 로봇은 벽과 장애물 사이에 두 개의 에지가 존재한다고 생각하게 된다. 이와 같은 에지를 퓨

전 에지(Fusion Edge)라고 정의하고자 한다. 그렇지만 이 퓨전 에지들은 실제로 에지가 아니다. 퓨전 에지들은 양 끝에 같은 노드들이 연결되어 있기 때문에 에지의 성질에 어긋나게 된다. 에지의 성질 중 에지들은 양 끝에 연결된 노드들이 다른 노드이어야 하기 때문이다. 로봇은 이 에지를 통해서 새로운 노드로 이동을 하려 하지만 새로운 노드가 나오지 않고 현재의 자신의 있는 노드라는 것을 인식하게 된다. 퓨전 에지는 서로의 시작 위치와 끝 위치가 바뀌어 있다는 특징을 가지고 위상학적 구조에서 제거할 수 있다.

Fig. 3(f)의 경우 에지는 한 쪽 끝이 막혀있는 경우이다. 이와 같은 에지를 데드 엔드 에지(Dead-End Edge)로 정의하고자 한다. 에지는 반드시 양쪽 끝이 노드에 연결되어야 한다는 성질이 있지만 데드 엔드 에지는 한쪽 끝이 연결되어 있지 않으므로 실제 에지라고 할 수 없다. 로봇은 이 에지를 통해 새로운 노드로 넘어가려 할 때 막혀 있다는 것을 알고 위상학적 지도에서 제거할 수 있다.

마지막으로 순환고리 경우(Cyclic Case)이다. 이 경우는 Fig. 3(a)와 Fig. 3(b)에서 나타나게 되는데 이 경우는 새로운 노드가 이전 노드 중에 부합되는 노드가 있는가를 확인해서 해결할 수 있다. 노드의 성질에서 노드들은 서로 독립된 공간들을 가지고 있다. 즉 서로 공간적으로 접할 수가 없다. 그러므로 이 노드가 이전에 왔었던 노드인가를 확인할 수가 있다. 만약 같은 노드가 이전에 있다면 새로운 노드로 추가하지 않고 현재의 에지를 이전에 있는 노드에서 찾아서 병합시키게 된다.

3.3 지도형성을 위한 자율주행 과정

앞에서 설명한 에지의 인식을 통한 생성 및 필요 없는 에지의 제거, 그리고 같은 에지 및 노드에 대해서는 통합과정을 통해서 위상학적 지도를 구성하게 된다. 전반적인 자율주행 방법은 다음과 같은 단계로 구성된다.

3.3.1 Pre-Mapping

이 과정에서 로봇은 기본적으로 벽을 따라 이동한다. 로봇이 벽을 따라 이동할 때 로봇은 에지를 인식하게 된다. 에지가 인식 되었을 때 에지의 방향에 수직인 방향으로 이동해야 한다. 그 래야 한 노드의 지도형성이 끝나기 전에 다른 노드로 넘어가지 않는다. Pre-Mapping이 끝나고 난

후에는 로봇은 에지가 될 수 있는 예상 에지(Candidate Edge)의 개수와 에지의 시작 위치, 노드의 전반적인 크기를 알 수 있다.

3.3.2 Full-Mapping

Full-Mapping은 노드의 전 영역에 대해 지도형성을 하는 과정이다. 본 연구에서는 지그제그(Zig-Zag) 알고리즘을 사용하였다. 그렇지만 지그제그 알고리즘은 약간은 비효율적인 알고리즘이다. Full-Mapping 과정에서 지그제그 알고리즘뿐만 아니라 다른 효율적인 알고리즘도 사용할 수 있다. 그렇지만 이 과정은 본 연구에서 중요 주제에 아니므로 더 이상 언급하지 않고자 한다.

3.3.3 Move to New Node

노드에서 Full-Mapping이 끝난 후 로봇은 현재의 노드에 가보지 않은 비탐색 에지(Unexplored Edge)가 존재하는지를 확인한다. 비탐색 에지가 존재하면 로봇은 비탐색 에지로 이동을 하게 된다. 이 때 이동 목표지점은 비탐색 에지의 시작 위치(Starting Position)이고 로봇은 A* 알고리즘으로 이동하게 된다. 로봇이 에지의 시작 위치에 도착하면 새로운 노드가 나타날 때까지 에지를 따라 이동하게 된다. 에지를 따라 이동을 할 때 먼저 새로운 노드가 나오는지 아니면 에지를 따라서 더 이상 이동할 수 없는지를 확인하는데 만약 로봇이 에지를 따라서 더 이상 이동할 수 없을 경우에는 이 에지는 Fig. 3(f)와 같이 한 쪽 끝에 다른 노드가 연결되지 않은 데드엔드 에지로 위상학적 지도에서 제거가 된다.

새로운 노드가 나타나면 먼저 Fig. 3(e)인 경우 인가를 확인한다. 에지를 따라서 이동 후 새로운 노드가 나타났을 때 그 위치가 에지의 끝 위치(Ending Position)가 되는데 끝 위치가 이 노드의 다른 에지의 시작 위치와 일치하게 된다면 이 경우는 퓨전 에지가 되고 두 에지들은 예상 에지에서 제거가 되어야 한다. 그리고 Fig. 3(e)의 경우가 아니라고 판단이 되었을 경우에는 새로운 노드가 이전 위상학적 지도에 존재하는 노드인가를 확인해야 한다. 이것은 순환고리경우인가를 확인하는 것이다. 가정환경 내에서 각 노드들은 위상학적 구조를 가지고 있지만 각 노드들은 상대적인 위치를 가지고 있고 이 위치들은 변화가 없으므로 각 노드들은 겹쳐서 존재할 수 없다. 그리

므로 상대적인 위치를 통해서 로봇은 이전에 나왔던 노드인지를 확인할 수 있게 된다.

이전에 나왔던 노드가 아닌 것이 판단되면 로봇은 새로운 노드를 위상학적 지도에 추가를 하고 로봇은 현재의 위치를 (0,0)으로 초기화한 후 새로운 로컬 그리드 맵을 형성하고 Pre-Mapping 과정부터 다시 시작하게 된다.

3.3.4 Move to Last Node

위상학적 구조를 구성하는 방법으로 본 연구에서는 깊이 우선 탐색(Depth First Search) 알고리즘을 사용하였다. 그러므로 비탐색 에지가 존재하지 않을 경우에 로봇은 상위 노드(Parent Node)로 이동을 한다. 이 과정에서 로봇은 상위 노드로 이동하는 에지를 찾아서 그 에지의 시작 위치로 이동하게 된다. 이동 알고리즘은 Move to New Node 과정에서 사용하였던 A*를 사용하였다. 상위 노드로 이동 후 로봇은 다시 비탐색 에지가 있는지를 확인하고 만약 있다면 Move to New Node 과정으로 돌아가 다시 새로운 노드를 찾는다. 상위 노드에도 비탐색 에지가 없으면 다시 상위 노드로 이동하게 된다. 상위 노드로 계속 이동 중에 더 이상의 상위 노드가 없으면 지도형성 과정은 끝나게 된다. 전체적인 자율주행 순서도는 Fig. 4와 같다.

3.4 가정환경에서 Topological Map의 장점

Topological Structure를 이용해 가정환경에서 지도를 형성 하였을 경우 다음과 같은 장점이 있다. 먼저 한 노드의 위치에서 다른 노드로 이동할 때 전체 지도를 검색해서 경로를 형성할 필요

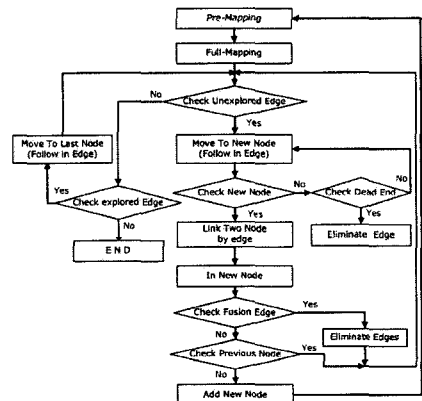


Fig. 4 Overall procedure of navigation

가 없고 현재 노드에서 가고자하는 노드까지 연결된 노드 지표(Node Index)들과 각 노드에서 다음 노드 지표를 가지고 있는 에지 위치까지의 경로들만 알면 된다. 이것은 계산적인 면에서 매우 효율적이다. 경로형성에서 전체 지도를 검색할 경우 짧게는 수초에서 수분까지 걸릴 수 있기 때문이다. 그렇지만 연결된 노드 지표 검색은 시간이 거의 걸리지 않고 각 노드에서 에지까지의 경로를 형성하는데 전체 지도를 검색하는 것보다 훨씬 적게 걸린다.

그리고 지도형성 시에도 한 노드의 지도형성을 끝내고 다른 노드로 이동을 해서 지도형성을 하기 때문에 효율적이다. 일반적인 가정환경에서 한 번에 전체 영역에 대해 지도형성을 시도하면 매우 복잡하고 비효율적이지만 한 노드를 끝내고 다른 노드로 이동해서 지도 형성을 하게 되면 같은 알고리즘을 적용한다고 해도 훨씬 간단하다.

마지막으로 각 노드에서 생긴 오차들은 서로 독립적이다. 각 노드로 이동을 했을 때마다 각 노드에 해당하는 지도를 사용하기 때문에 각 노드에서 생긴 오차들은 각 노드에 한정적으로 영향을 미칠 수밖에 없다. 이것은 지도형성시 오차가 커지는 것을 막아주게 된다.

위와 같은 이유 때문에 가정환경에서 위상학적 구조를 가진다는 것이 커다란 장점이라 할 수 있겠다.

4. 실험 및 고찰

4.1 위치인식 실험

2장에서 설명한 바코드를 이용한 위치인식이 잘 되는지 검증하기 위해서 두 가지 실험을 하였다. 실험은 엔코더 성능이 좋지 않은 로봇(300pul./rev.)인 Fig. 1(b)을 이용해 실험 하였다.

첫 번째 실험은 로봇이 1.5m × 1.5m의 사각형 경로를 11번 회전하게 한 후 로봇의 최종위치에서 순수한 오도메트리 정보와 오도메트리와 바코드를 이용해서 보정한 위치가 실제 위치와 얼마나 비교하였다. 로봇의 속도는 0.15m/s이고 총 이동거리는 66m이다. 회전 후 처음 위치에서 얼마나 벗어났는지 확인하였다. Fig. 5(a)에서 빨간선은 오도메트리 정보이고 파란색은 바코드와 오도메트리를 가지고 보정한 정보이다. Fig. 5(a)에서 보듯이 오도메트리 정보는 계속해서 발산하려하

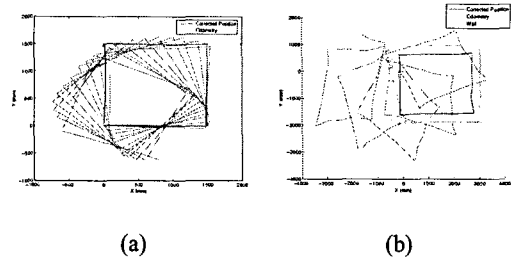


Fig. 5 Experiment results of localization

Table 2 Final robot position

	$x_{position}(cm)$	$y_{position}(cm)$
Real Position	1.0	3.6
Odometry	79.1	-62.3
Corrected Position	0.1	0.2
Odometry error	78.1	-65.9
Corrected Position Error	-0.9	-3.4

고 보정된 정보는 정확하다는 것을 알 수 있다. Table 2는 로봇의 최종 위치에서의 정보이다. 실제위치에 대해 오도메트리의 최종위치는 (78.1, -65.9)cm의 오차를 갖지만 보정된 최종위치는 (-0.9, -3.4)cm의 오차를 갖는다. 보정된 로봇의 위치 오차는 로봇의 중심점과 바코드 중심점 사이의 거리에 따른 오차로 dcx 과 dcy 을 정확하게 보정해 줄수록 오차는 줄어들게 된다. 그렇지만 이 오차들은 바코드 간격보다는 클 수가 없으므로 dcx 과 dcy 의 정확한 오차보정이 꼭 필요한 것은 아니다. 현재 바코드가 3.75cm 간격으로 배열이 되어 있으므로 로봇이 갖는 오차는 3.75cm를 넘을 수 없게 된다.

두 번째 실험은 로봇이 주어진 경로를 정확하게 따라갔는지 알아보기 위해 3.4m × 2.7m방에서 로봇은 벽을 따라 5바퀴를 회전하도록 하고 0.15cm/s 속도로 이동하였고 Fig. 5(b)는 로봇의 회전이 끝난 후의 결과이다. 빨간색은 오도메트리 정보이고 파란색은 보정된 로봇의 위치, 그리고 노란색은 보정된 위치와 그 때의 센서의 거리를 이용해서 벽의 위치를 표현한 것이다. 만약 보정된 위치가 부정확하다면 벽의 모양은 정확하게 사각형 모양이 나오지 않고 많이 틀어진 모습이 나와야 한다. 그렇지만 벽이 사각형 모양을 정확하게 나타내고 있음을 알 수 있고 순수한 오도메트리 정보는 발산함을 알 수 있다. 위 두 번의 실험을 통해서 2장에서 설명한 위치인식 방법이 잘 되고 있음을 알 수 있다.

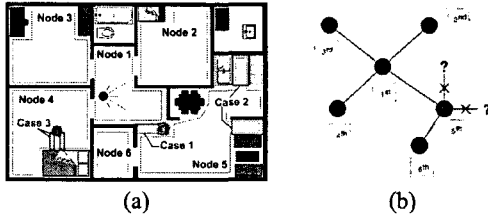


Fig. 6 Indoor environment in simulation

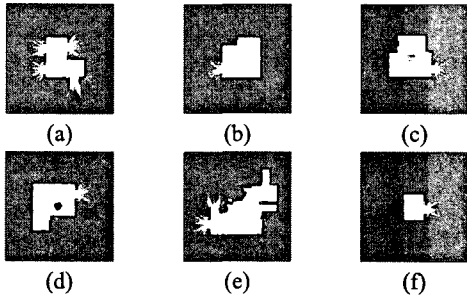


Fig. 7 Local grid map

4.2 모의실험 결과

가정환경에서 위상학적 구조를 구성을 위한 자율주행 방법을 보이기 위해 Fig. 6(a)와 같은 환경에서 모의실험을 먼저 수행하였다. 모의실험 환경을 실제 환경과 유사하기 하기 위해 로봇의 반지름은 210mm, 가정환경의 크기는 8m×12m로 하고 벽의 두께는 100mm로 해서 Fig. 6(a)처럼 축소화 시켜서 모의실험을 수행했다. Fig. 6(a)와 같은 환경은 Fig. 6(b)처럼 크게 6개의 노드로 나뉘게 된다. Fig. 6(a)에서 Case 1은 Fig. 3(c)의 경우라고 할 수 있겠다. 에지가 벽과 화분에 의해 생긴 경우이다. Case 2는 Fig. 3(f)처럼 테드 엔드 에지이고 위상학적 지도에서 제거한다. Case 3은 Fig. 3(e)과 같은 퓨전에지이다. 모의실험 결과로 나온 각 노드들의 로컬 그리드 맵 정보들은 Fig. 7과 같다. 각 노드에 연결된 에지의 개수는 1번 노드가 4개, 5번 노드가 2개 2, 3, 4, 6번 노드가 1개씩이다. 결과에서 Case 2와 Case 3은 에지로 취급되지 않았음을 알 수 있다. 로봇이 이동한 노드 과정은 다음과 같다.

Node 1 → Node 2 → Node 1 → Node 3 → Node 1 → Node 4 → Node 1 → Node 5 → Node 6 → Node 5 → Node 1

4.3 자율주행 실험 및 결과

앞에서 제안한 위상학적 정보를 이용한 자율주행을 검증하기 위해 다양한 환경에서 실험을 수

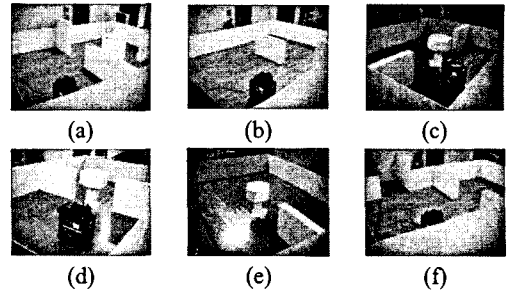


Fig. 8 Environments of experiment

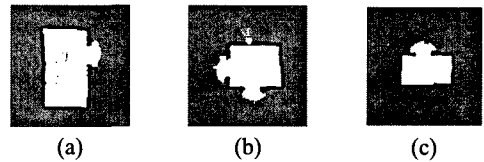


Fig. 9 Topological structure of environments

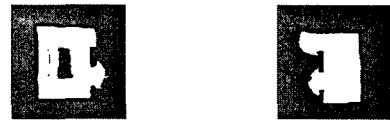


Fig. 10 Local grid map in Case 1

행하였다(Fig. 8). 실험은 바코드가 있는 바닥재가 깔려있는 환경에서 수행되었고 로봇은 Fig. 1(c)의 BSR I을 사용하였다. 로봇은 모르는 환경을 스스로 자율주행 하면서 맵을 얻었다.

4.3.1 위상학적 지도(Topological Map) 구성

먼저 간단히 3개의 노드로 구성이 되어 있는 Fig. 8(a)환경에서 실험을 하였다. 로봇은 먼저 첫 번째 노드를 지도형성을 하며 지도형성이 끝나고 나면 비탐색 에지(Unexplored Edge)가 하나 존재하는 것을 알기 때문에 그 에지를 통해 새로운 노드로 이동한다. 이와 같은 과정으로 세 번째 노드까지 지도형성을 한 후 다시 첫 번째 노드로 돌아왔을 때 전체 지도형성 과정을 끝낸다. 이 지도형성 과정에서 얻어진 각 노드의 로컬 그리드 맵은 Fig. 9와 같다.

4.3.2 case 1의 실험결과[Fig. 3(a)]

Fig. 3(a)에서 설명하였듯이 가정환경에는 방과 방 사이를 연결하는 에지들이 Fig. 8(b)처럼 여러 개 존재할 수 있다. Fig. 8(b)는 두 영역 사이에서 서로의 영역으로 넘어갈 수 있는 문과 같은 영역이 두 개가 있는데 이와 같은 경우의 위상학적

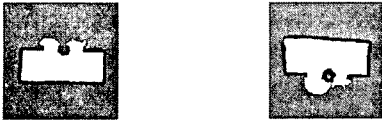


Fig. 11 Local grid map in Case 2



Fig. 12 Local grid map in Case 4 and Case 5



Fig. 13 Local grid map in two node and one dead-end edge environment

구조는 순환고리(Cyclic loop)가 된다. 즉 1번 노드에 다시 왔을 때 1번 노드라는 것을 알지 못한다면 3번 노드로 추가하게 되고 계속해서 무한순환에 빠지게 되는 것이다. 그렇지만 앞에서 설명하였듯이 각 노드들은 독립된 공간들을 가지고 있다는 성질을 가지고 있으므로 로봇이 1번 노드로 왔을 때 로봇은 새로운 노드가 아니라 예전에 지도형성을 했던 1번 노드라는 것을 알게 되고 새로운 노드를 추가하지 않는다. 그리고 현재의 에지가 1번 노드에서 어느 에지와 같은지를 검색해서 정보를 갱신한다. 실험 후 각 노드들의 로컬 그리드 맵은 Fig. 10과 같다.

4.3.3 Case 2의 실험결과[Fig. 3(b)]

Fig. 3(b)는 Fig. 3(a)와 같이 에지가 2개 있는 경우이다. Fig. 8(c)처럼 원래는 에지가 한 개였지만 사이에 장애물이 놓여 있어서 에지가 2개인 경우로 바뀌게 된 것이다. 위상학적 구조는 앞에서 설명한 Case 1과 같은 구조이다. 실험 결과로 얻어진 각 노드의 로컬 그리드 맵은 Fig. 11과 같다.

4.3.4 Case 3의 실험결과[Fig. 3(c)]

Case 2경우처럼 에지 사이에 장애물이 놓여 있는 경우지만 로봇은 에지가 하나라고 인식하는 경우이다. 이 경우에 기존의 에지의 위치가 약간

의 변동이 있다는 것 외에는 변화가 없다. 그러므로 인식된 에지로 로봇은 이동할 수 있다.

4.3.5 Case 4의 실험결과[Fig. 3(d)]

Case 4는 Fig. 3(d)에서 설명했던 것처럼 로봇이 장애물에 의해 막혀서 로봇이 다른 노드로 이동하지 못하는 경우이다. Fig. 8(d)에서는 노드가 2개가 존재하고 노드 사이에는 연결하는 통로가 하나가 있지만 그 통로 사이에 장애물이 놓여 있어서 로봇은 다른 노드로 건너 갈 수 없다. 이것은 가정에 어긋나는 것으로 이와 같은 경우는 지도형성시 제외되어야 한다. 첫 번째 노드의 지도형성 결과는 Fig. 12(a)와 같다.

4.3.6 Case 5의 실험결과[Fig. 3(e)]

Fig. 8(e)에서 벽과 장애물 사이에 퓨전에지가 존재한다. Pre-Mapping이 끝난 후 로봇은 두 개의 예상에지를 갖는데 이것은 퓨전에지들이다. 로봇은 예상에지를 통해 다른 노드로 넘어가려 할 때 이 예상에지들이 퓨전에지임을 알게 되고 위상학적 구조에서 제거한다. 지도형성 결과는 Fig. 12(b)와 같다.

4.3.7 Case 6의 실험결과[Fig. 3(f)]

Fig. 8(f)는 노드 2개와 2개의 노드를 연결하는 에지, 그리고 첫 번째 노드에 연결된 데드 엔드 에지가 있다. 로봇은 첫 번째 노드 지도형성 후 에지가 두 개 있다고 인식을 하고 그 중에서 첫 번째 에지를 통해서 두 번째 노드로 넘어간다. 두 번째 노드의 지도형성 후에는 비탐색 에지가 없는 것을 알게 되고 상위 노드인 첫 번째 노드로 넘어온다. 첫 번째 노드에서 비탐색 에지인 두 번째 에지로 이동해서 새로운 노드로 넘어가려 하지만 두 번째 에지는 데드 엔드 에지로 다른 노드로 연결되지 않았다. 이것은 에지의 성질에 맞지 않으므로 이 에지를 위상학적 구조에서 제거하게 된다. 지도형성 결과는 Fig. 13과 같다.

4.4 키드넵(Kidnap) 상황 극복

위와 같은 과정으로 지도형성 후에는 전 영역에 대한 바코드 정보를 갖게 된다. 그러므로 키드넵(Kidnap)된 상황을 극복할 수 있다. 그렇지만 같은 값을 갖는 바코드는 집안 여러 곳에 존재할 수 있다. 읽힌 바코드와 같은 값이 지도상에 어디에 존재하는지를 확인해야 한다. 존재하는 위

치가 한 곳에만 있을 경우에는 바로 키드넵 상황이 극복되지만 만약 여러 곳에 존재할 경우에는 로봇이 이동을 하면서 각 예측되는 위치들을 추정 하면서 가능한 위치들을 추정해 나간다. 예측되는 위치가 한 곳으로 좁혀질 경우 그 위치가 키드넵 상황을 극복한 위치가 되는 것이다. 실험을 통해 키드넵 상황 극복을 검증하였다.

5. 결론

이 논문은 눈에 보이지 않는 바코드를 이용하여 가정환경에서 위치인식과 위상학적 구조를 갖는 자율주행 방법을 제시하였다. 제안된 알고리즘들은 다음과 같은 특징을 갖는다.

먼저 눈에 보이지 않는 바코드를 이용하여 위치인식 문제를 풀었다. 이 방법은 수 센티미터의 에러만을 가질 뿐만 아니라 전체 지도 형성 후에는 키드넵 상황도 극복할 수 있다는 장점이 있고 다른 위치인식 알고리즘들에 비해 계산속도가 빠르다. 그리고 바코드가 손실되거나 잘못된 바코드 정보가 들어오는 경우도 극복이 가능하다.

두 번째로 새로운 노드와 에지에 대해 정의를 함으로써 여러 개의 방들을 가지고 있는 가정환경에 적합한 자율주행 알고리즘을 제안하였다. 가정환경에 적합한 위상학적 구조를 제안함으로써 지도형성을 할 때 한 영역의 지도형성을 끝내고 다른 영역으로 넘어가므로 쉽게 지도형성 알고리즘을 적용할 수 있고 각 노드에서 생긴 오차들은 서로 독립적이므로 오차가 계속적으로 증가하는 것을 막을 수 있다. 그리고 경로형성을 할 때에도 계산시간 면에서 많은 이득이 있다.

전 과정들을 모의실험과 실제 환경에서의 실험을 통해 검증 및 평가를 하였다.

후 기

이 연구는 한국 과학기술부의 국가지정연구실 사업(M1-0302-00-0040-03-J00-00-024-00)과 정보통신부의 URC를 위한 내장형 컴포넌트 기술개발 및 표준화 과제 (A1100-0602-0026) 지원을 받아 수행되었습니다.

참고문헌

(1) iRobot. Website. [Online]. Available: <http://www.>

roombavac.com

(2) Electrolux. Website. [Online]. Available: <http://trilobite.electrolux.com/>

(3) Dellaert F., Fox D., Burgard W. and S. Thrun, 1999, "Monte Carlo Localization for Mobile Robots," *Proc. IEEE International Conference on Robotics and Automation*, pp. 1322~1328.

(4) Gonz'alez-Banos H. H. and Latombe J.-C., 2002, "Navigation Strategies for Exploring Indoor Environments," *International Journal of Robotics Research*, Vol. 21, No. 10, pp. 829~848.

(5) Enrique Gonzalez, Oscar Alvarez, Yul Diaz, Carlos Parra and Cesar Bustacara, 2005, "BSA: A Complete Coverage Algorithm," *Proc. IEEE International Conference on Robotics and Automation*, pp. 2052~2056.

(6) Huang W. H., 2001, "Optimal Line-sweep-based Decompositions for Coverage Algorithms," *Proc. IEEE International Conference on Robotics and Automation*, pp. 27~32.

(7) Tomatis N., Nourbakhsh I., and Siegwart R., 2001, "Simultaneous Localization and Map Building: A Global Topological Model with Local Metric Maps," *Proc. IEEE International Conference on Intelligent Robots and Systems*, pp. 421~426.

(8) Bosse M., Newman P., Leonard J., Soika M., Feiten W. and Teller S., 2003, "An Atlas Framework for Scalable Mapping," *Proc. IEEE International Conference on Robotics and Automation*, pp. 1899~1906.

(9) Victorino A. C. and Rives P., 2005, "Global Consistency Mapping with an Hybrid Representation," *Proc. IEEE International Conference on Intelligent Robots and Systems*, pp. 2403~2408.

(10) Madsen C. and Andersen C., 1998, "Optimal Landmark Selection for Triangulation of Robot Position," *International Journal of Robotics and Autonomous Systems*, Vol. 23, No. 4, pp.277~292.

(11) Hern'andez S., J. Torres M., C. Morales A., and Acosta L., 2003, "A New Low Cost System for Autonomous Robot Heading and Position Localization in a Closed Area," *Autonomous robots*, Vol. 15, No. 2, pp. 99~110.