

온톨로지에 기반한 자율주행 로봇의 운항

Ontology-based Navigational Planning for Autonomous Robots

이인근, 서석태, 정혜천, 권순학*

In K. Lee, Suk T. Seo, Hye C. Jeong, Soon H. Kwon*

영남대학교 전기공학과

Dept. of Electrical Engineering, Yeungnam University

요 약

인간의 지속적인 도움 없이 거칠고 변화하며 구조화되지 않은 환경에서 원하는 작업을 수행하는 자율주행 로봇은 확실하거나 불확실한 주변 환경을 극복하는 능력을 지녀야 한다. 이를 위해서는 센서로부터 얻어진 불확실한 정보를 바탕으로 유용한 결론을 도출하는 알고리즘이 요구된다. 본 논문에서는 지식의 표현 및 처리에 유용한 방법으로 주목을 받고 있는 온톨로지에 기반한 자율주행 로봇의 자율주행 알고리즘을 제안하고 이의 타당성을 컴퓨터 모의실험을 통해 보인다.

키워드 : 자율주행 로봇, 온톨로지, 퍼지 논리, 자율주행

Abstract

Autonomous robots performing desired tasks in rough, changing, unstructured environments without continuous human assistance must have the ability to cope with its surroundings whether this be certain or not. The development of algorithms deriving useful conclusions from uncertain information obtained by various sensors may be the first for it. Recently ontology is taken great attention as a method useful for the representation and processing of knowledge. In this paper, we propose an ontology-based navigation algorithm for autonomous robots, and provide computer simulation results in order to show the validity of the proposed algorithm.

Key Words : Autonomous robots, Ontology, Fuzzy logic, Autonomous navigation

1. 서 론

지능형 로봇에 대한 관심이 증대되면서 로봇의 지능 구현 방법에 관한 연구가 진행되고 있다. 로봇의 지능 구현을 위해 퍼지 논리, 신경망, 유전자 알고리즘과 같은 지능적 알고리즘이 이용되고 있으며, 이들을 이용하여 로봇 주변의 상황을 판단하고 주어진 목표를 수행하는 자율주행 로봇의 연구가 활발히 진행 중이다[1,2,6-9].

로봇의 자율주행에 관한 기존의 연구[6-9]에서는 수치적으로 표현된 좌표 기반의 지도로부터 목표점까지의 최단 경로를 알아내고, 국소 지역에서 로봇의 자율주행을 위한 장애물의 회피에 관한 연구가 대부분이다. 즉, 수치적으로 표현된 좌표 지도를 이용하여 목표점의 정확한 위치와 장애물의 상세한 정보를 알 수 있으나, 로봇이 주행하는 과정에서 획득한 정확하지 않은 상황 정보로 인한 잘못된 판단에 유동적으로 대처하기 어려운 문제가 있다.

지도의 정보를 수치적으로 표현하지 않고, 장애물의 속성과 장애물간의 관계정보만을 표현한 '언어 지도(linguistic map)'를 이용한 방법이 제안되었다[1,2]. 이 연구에서는 '인지(cognition)'와 '지각(perception)'에 기반한 지능형 자율주

행 로봇의 구현에 중점을 두었으며, 불확실한 환경정보와 상황정보에서도 효율적으로 로봇의 위치를 판단하고 목표점까지 이동하는 방법이 제안되었다. 그러나 정형화 되어 있는 '언어 지도'는 확장성이 낮고, 로봇의 위치 판단을 위한 '퍼지 추론'은 추론 규칙의 개수에 따라 연산량이 매우 커지는 단점이 있다.

최근에는 인간과 기계가 서로 지식을 공유하고 표현하는 방법으로 온톨로지를 이용하는 연구가 주목을 받고 있다. 온톨로지는 개념(concept) 및 개체(instance)를 정의하고, 개념 및 개체의 특징(attribute)과 개념간의 관계(relation), 개체간의 관계를 다양하게 표현할 수 있다. 또한 온톨로지 추론을 통해 온톨로지서 표현되지 않은 새로운 관계를 생성할 수 있는 장점이 있다[3]. 즉, 온톨로지를 이용하여 로봇의 주행 환경 정보를 자유롭게 표현하고 로봇의 상황 변화에 따라 새로운 사실을 추론함으로써 지식 정보의 효용성을 높일 수 있다.

본 논문에서는 로봇이 이동할 환경과 로봇이 감지한 장애물의 상황 정보를 온톨로지서 표현함으로써 정보의 표현 및 확장이 용이하도록 하고, 온톨로지 추론과 검색을 통해 로봇의 위치를 판단하는 온톨로지에 기반한 로봇의 자율주행 알고리즘을 제안한다. 그리고 컴퓨터 모의실험을 통해 제안된 알고리즘의 타당성을 보인다.

접수일자 : 2007년 7월 12일

완료일자 : 2007년 8월 6일

*Corresponding author

2. 온톨로지에 기반한 자율주행 로봇의 운항

본 논문에서 제안하는 온톨로지 기반의 자율주행 알고리즘은 그림 1과 같다. 제안하는 알고리즘은 로봇이 주행할 환경의 지도를 온톨로지 구성하고, 로봇이 감지한 장애물 정보를 온톨로지 표현한다. 그리고 온톨로지 추론과 온톨로지 검색을 통해 로봇의 위치를 판단하고, 목표점까지의 경로를 찾아 이동한다.

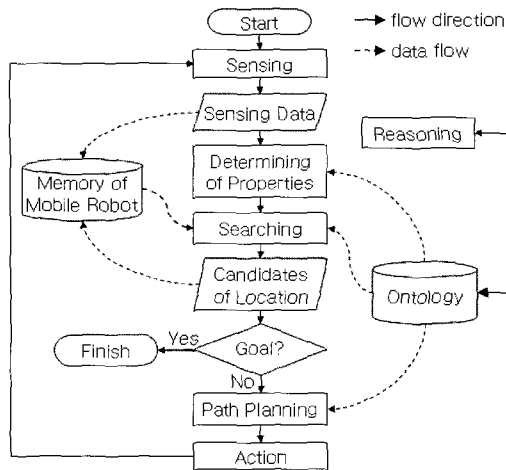


그림 1. 온톨로지 기반 자율주행 알고리즘
Fig. 1. Ontology-based autonomous navigation algorithm

2.1 지식 베이스

2.1.1 지도 온톨로지

본 논문에서는 로봇의 주행 공간의 환경 정보를 표현한 온톨로지를 “지도 온톨로지(map-ontology)”라고 정의한다.

지도 온톨로지를 구성하기 위한 방법은 그림 2와 같다. 그림 2(a)의 환경 정보를 그림 2(b)와 같이 온톨로지 표현한다. 로봇의 이동을 방해하는 것은 모두 “장애물(Obstacle)”로 간주한다. “장애물”은 “무형(Abstract) 장애물”과 “유형(Material) 장애물”로 구분한다. “무형 장애물”에는 신호등과 같은 “시간(Temporal) 장애물”과 맞은편 차도와 같은 “공간(Spatial) 장애물”로 구분한다. “유형 장애물”에는 모바일 로봇과 같은 이동 능력을 가진 “이동(Active) 장애물”과 건물과 같이 이동 능력이 없는 “고정(Static) 장애물”로 구분한다. 그리고 실존하는 장애물 ‘A’, ‘B’, ‘C’, ‘D’는 “고정 장애물”의 개체로 표현된다. 2차원 지도에는 나타나지 않았지만 이동 능력이 있는 로봇은 “이동 장애물”의 개체로 표현된다.

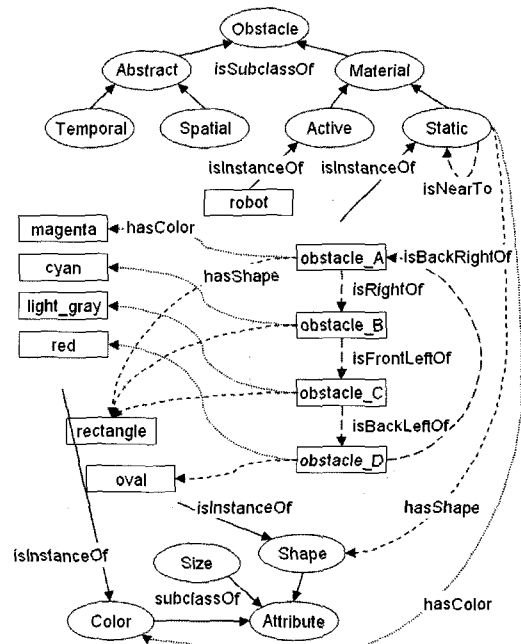
장애물의 특징은 “색상(Color)”, “모양(Shape)”, “크기(Size)” 등으로 표현하였다. 그림 2(b)에서 “색상”의 개체로 표현되는 ‘magenta’, ‘cyan’, ‘light_gray’, ‘red’와 “모양”의 개체로 표현되는 ‘rectangle’, ‘oval’은 온톨로지 개념으로 표현할 수도 있다. 그러나 본 논문에서는 이들을 장애물의 특징을 표현하는 최소 단위로 간주하여 개체로 표현하였다.

지도 온톨로지를 구성함에 있어 개념 간, 개체 간 관계를 표현하기 위한 “속성(Property)”은 그림 3과 같다. 최상위 속성인 “Property”의 하위속성인 “hasCharacter”는 장애물의 특징을 표현하기 위한 속성이며, “hasColor”, “hasShape”, “hasSize”를 하위속성으로 가진다. “isNearTo”는 장애물 사

이의 위치 관계를 표현하기 위한 속성으로 8방향(FT(front), FR(front-right), RT(right), BR(back-right), BK(back), BL(back-left), LT(left), FL(front-left))을 나타낸다. 그림 2(a)의 장애물 ‘A’와 ‘D’의 관계에서 장애물 ‘D’는 장애물 ‘A’의 ‘back-right’에 위치하므로 “isBackRightOf(obstacle_D, obstacle_A)”로 표현한다. 반대로 장애물 ‘A’는 장애물 ‘D’의 ‘front-left’에 위치하므로 “isFrontLeftOf(obstacle_A, obstacle_D)”로 표현한다. 그러므로 ‘isBackRightOf’와 ‘isFrontLeftOf’는 반대 의미의 속성이다.

Obstacle	Color value			Color Attribute
	Red	Green	Blue	
A	230	50	230	Magenta
B	40	240	240	Cyan
C	190	195	183	light gray
D	230	10	15	red

(a) 장애물의 배치와 특징



(b) 지도 온톨로지

그림 2. 장애물의 온톨로지 표현
Fig. 2. Ontological representation of obstacles

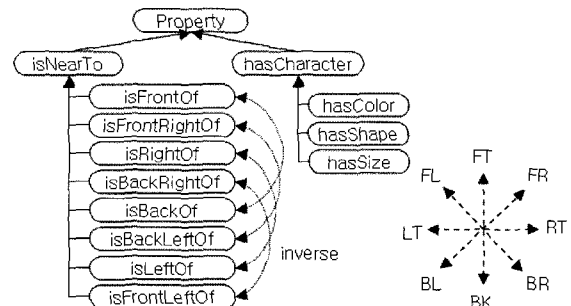


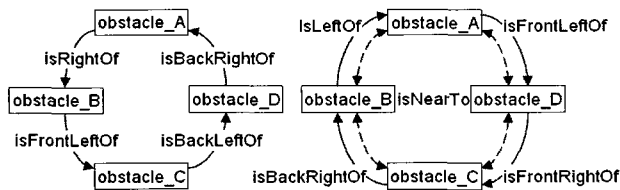
그림 3. 지도 온톨로지에서의 속성
Fig. 3. Properties on map-ontology

2.1.2 온톨로지 추론

환경 정보를 온톨로지로 쉽게 표현하고 수정할 수 있다. 그러나 세세한 정보까지 일일이 온톨로지로 표현하기에는 너무 많은 시간과 노력이 소요된다. 따라서 중요한 환경 정보만을 온톨로지로 표현하고, 온톨로지 추론을 통해 상세한 환경 정보를 정의하여 지도 온톨로지를 확장한다.

온톨로지 추론은 SWRL(Semantic Web Rule Language)[4]을 이용한다. SWRL은 기존 정보를 바탕으로 새로운 정보를 도출하기 위한 지식의 표현 방법으로 W3C에서 제안하였다. 다음은 SWRL을 이용한 온톨로지 추론의 예를 보인다. 그림 2에서 “isRightOf(obstacle_A, obstacle_B)”의 정보만이 표현되었다. 그러나 표 1의 ‘rule-3.2’에 의해 “isNearTo(obstacle_A, obstacle_B)”이라는 새로운 관계가 추론된다.

그림 4는 그림 2(b)의 지도 온톨로지를 표 1의 규칙으로 추론하여 새로운 사실을 도출하는 과정을 보인다.



(a)추론 전의 개체 관계 (b)추론된 개체 관계
그림 4. 온톨로지 추론

Fig. 4. Ontological inference

2.2 감지

본 논문에서는 레이저 변위센서와 카메라를 이용하여 로봇과 장애물 사이의 거리를 측정하고 로봇 주변 장애물의 색상을 감지한다. 로봇과 장애물의 배치가 그림 5(a)와 같을 때 로봇 주변 장애물의 카메라 영상은 그림 5(b)와 같다. 그리고 그림 5(a)에서 로봇의 감지 범위 내의 카메라 영상은 그림 5(c)와 같다.

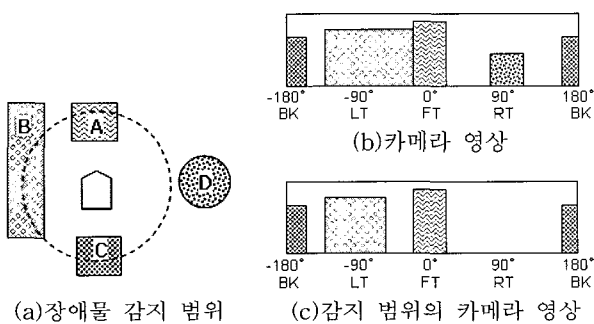


그림 5. 로봇의 장애물 감지
Fig. 5. Obstacle detection of a robot

2.3 획득

로봇이 감지한 장애물의 정보를 온톨로지로 표현함으로써 개체의 속성뿐만 아니라 개체 사이의 관계도 표현할 수 있다. 그리고 로봇의 주변 상황을 온톨로지로 표현한 “상황 온톨로지(Circumstance-ontology)”로부터 지도 온톨로지를 검색함으로써 로봇의 위치를 쉽게 판단할 수 있다. 그림 6은 로봇이 그림 5(a)에서 장애물을 감지하였을 때, 감지한 장애물 정보

표 1. 추론 규칙
Table 1. Inference rules

번호	규칙
rule-1.1	$isFrontOf(?x, ?y) \wedge Static(?x) \wedge Static(?y) \rightarrow isBackOf(?y, ?x)$
rule-1.2	$isFrontOf(?x, ?y) \wedge Static(?x) \wedge Static(?y) \rightarrow isNearTo(?x, ?y)$
rule-2.1	$isFrontRightOf(?x, ?y) \wedge Static(?x) \wedge Static(?y) \rightarrow isBackLeftOf(?y, ?x)$
rule-2.2	$isFrontRightOf(?x, ?y) \wedge Static(?x) \wedge Static(?y) \rightarrow isNearTo(?x, ?y)$
rule-3.1	$isRightOf(?x, ?y) \wedge Static(?x) \wedge Static(?y) \rightarrow isLeftOf(?y, ?x)$
rule-3.2	$isRightOf(?x, ?y) \wedge Static(?x) \wedge Static(?y) \rightarrow isNearTo(?x, ?y)$
rule-4.1	$isBackRightOf(?x, ?y) \wedge Static(?x) \wedge Static(?y) \rightarrow isFrontLeftOf(?y, ?x)$
rule-4.2	$isBackRightOf(?x, ?y) \wedge Static(?x) \wedge Static(?y) \rightarrow isNearTo(?x, ?y)$
rule-5.1	$isBackOf(?x, ?y) \wedge Static(?x) \wedge Static(?y) \rightarrow isFrontOf(?y, ?x)$
rule-5.2	$isBackOf(?x, ?y) \wedge Static(?x) \wedge Static(?y) \rightarrow isNearTo(?x, ?y)$
rule-6.1	$isBackLeftOf(?x, ?y) \wedge Static(?x) \wedge Static(?y) \rightarrow isFrontRightOf(?y, ?x)$
rule-6.2	$isBackLeftOf(?x, ?y) \wedge Static(?x) \wedge Static(?y) \rightarrow isNearTo(?x, ?y)$
rule-7.1	$isLeftOf(?x, ?y) \wedge Static(?x) \wedge Static(?y) \rightarrow isRightOf(?y, ?x)$
rule-7.2	$isLeftOf(?x, ?y) \wedge Static(?x) \wedge Static(?y) \rightarrow isNearTo(?x, ?y)$
rule-8.1	$isFrontLeftOf(?x, ?y) \wedge Static(?x) \wedge Static(?y) \rightarrow isBackRightOf(?y, ?x)$
rule-8.2	$isFrontLeftOf(?x, ?y) \wedge Static(?x) \wedge Static(?y) \rightarrow isNearTo(?x, ?y)$

와 로봇과 장애물의 관계를 온톨로지로 표현한 것이다.

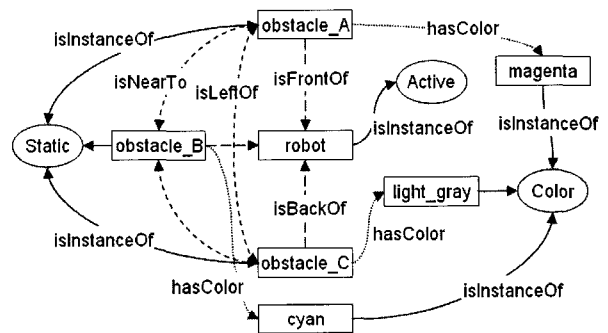


그림 6. 상황 온톨로지
Fig. 6. Circumstance-ontology

로봇이 감지한 장애물 개체인 ‘obstacle_A’는 로봇의 전방(front)에 위치하므로 “isFrontOf(obstacle_A, robot)”으로 표현한다. 그리고 ‘obstacle_B’는 로봇의 ‘front-left’, ‘left’, ‘back-left’의 3방향에 존재한다고 판단할 수 있다. 따라서 장애물의 방향은 식 (1)과 식 (2), 그리고 그림 7의 소속 함수 값에 의해 결정한다. 식 (1)에서 φ_{start} 는 특정 장애물의 감지가 시작된 각도이고, φ_{end} 는 감지가 끝난 각도이다.

$$\varphi_{mean} = C + \frac{\varphi_{start} + \varphi_{end}}{2} \quad (1)$$

$$C = \begin{cases} 0^\circ & (\varphi_{start} < \varphi_{end}) \\ 180^\circ & (\varphi_{start} \geq \varphi_{end}) \end{cases}$$

$$object\ direction = \arg \max_D (\mu_D(\varphi_{mean})) \quad (2)$$

여기서 D 는 방향에 대한 소속 함수이고, $\mu_D(\varphi)$ 는 각도 φ 에서 방향에 대한 소속 함수 값이다. 예를 들면, 그림 6(a)의 장애물 'C'의 경우, 각도가 $\varphi_{start} = 165^\circ$, $\varphi_{end} = -170^\circ$ 이고 $\varphi_{mean} = 177.5^\circ$ 이므로 장애물 'C'의 방향은 'BT' 이다.

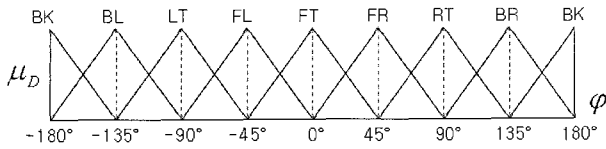


그림 7. 8방향의 소속 함수

장애물의 RGB의 각 색상 값을 8 bit로 표현하였을 때 장애물의 색상을 11가지(red, yellow, green, cyan, blue, magenta, red, black, dark-gray, gray, light-gray, white)로 나타내었다. 장애물의 색상은 식 3과 그림 8의 소속 함수 값에 따라 결정한다. 장애물 'A'의 색상 값이 ($z_R:100$, $z_G:200$, $z_B:250$)인 경우, 장애물 'A'의 색상은 'magenta'이다.

$$object\ color = \arg \max_C (\mu_C) \quad (3)$$

$$\mu_C = t_{ap}(\mu_C(z_R), \mu_C(z_G), \mu_C(z_B))$$

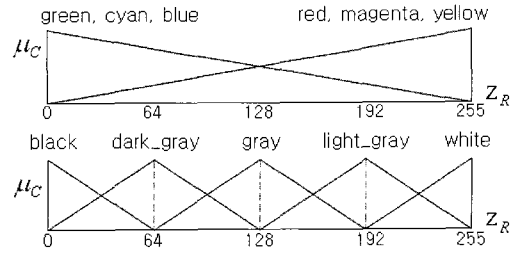
여기서 C 는 색상 소속 함수, $\mu_C(z)$ 는 색상 값 z 에서의 소속 함수 값이다. $t_{ap}()$ 는 대수곱(algebraic product)이다.

2.4 지각

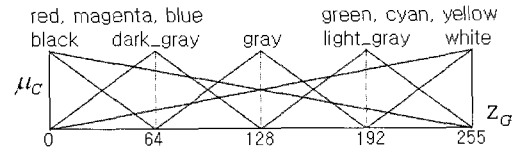
2.4.1 위치 판단을 위한 온톨로지 검색

상황 온톨로지에서부터 검색 질의를 생성하고, 지도 온톨로지를 검색하여 이동 로봇의 현재 위치를 판단한다. 이를 위해 로봇이 감지한 장애물들과 동일한 형태의 장애물의 특징과 관계정보를 지도 온톨로지에서부터 검색하여 로봇의 현재 위치 후보를 생성한다. 온톨로지 검색에는 SPARQL (Simple Protocol and RDF Query Language)[5]을 이용한다.

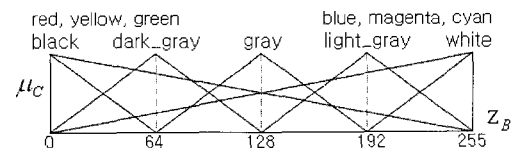
예를 들어 그림 9에서 'A'와 'D'의 색상이 'magenta', 'C'와 'E'의 색상이 'cyan'으로 특징이 동일하다. 그리고 'A'-'B'-'C'의 위치와 'D'-'B'-'E'의 위치가 'B'를 중심으로 대칭적으로 위치한다. 그림 9(a)에서 획득한 장애물 정보로부터 지도 온톨로지에서도 표 2의 질의를 통해 동일한 형태의 장애물을 검색하면 'A'-'B'-'C'와 'D'-'B'-'E'가 검색된다. 이들이 로봇의 위치 후보가 된다. 즉, 그림 9(a)의 위치에서 획득한 장애물의 정보에 의하면 로봇이 그림 9(a)의 위치뿐만 아니라 그림 9(b)의 위치에도 존재한다고 판단할 수 있다. 그러므로 지도 온톨로지에서도 동일한 형태의 장애물 특징과 관계 정보를 검색하여 위치 후보를 선택한다.



(a)'red'값의 소속 함수



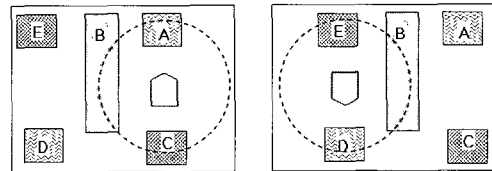
(b)'green'값의 소속 함수



(c)'blue'값의 소속 함수

그림 8. 색상의 소속 함수

Fig. 8. Membership functions for color



(a)로봇의 위치 1 (b)로봇의 위치 2

그림 9. 동일한 패턴의 장애물 위치

Fig. 9. Position of obstacles on same patterns

표 2. 장애물 검색을 위한 SPARQL 질의

Table 2. A SPARQL query for searching for obstacles

Query	<pre>PREFIX isac: <http://isac.yu.ac.kr#> SELECT ?x, ?y, ?z WHERE { ?x isac:isNearTo ?y . ?y isac:isNearTo ?z . ?z isac:isNearTo ?x . ?x isac:hasColor isac:magenta . ?y isac:hasColor isac:cyan . ?z isac:hasColor isac:light_gray . }</pre>
Results	<pre>x : y : z = A : B : C x : y : z = D : B : E</pre>

2.4.2 애매성의 축소

로봇의 현재 위치의 후보가 다수 존재하는 것은 로봇이 획득한 장애물 정보와 지도 온톨로지에 애매성이 포함되어 있기 때문이다. 따라서 참고문헌 [1]과 같이 애매성이 포함된 정보로부터 로봇의 현재 위치를 판단하기 위해 이전에 획득한 장애물 정보를 이용한다. 그리고 과거에 획득한 장애물 정보로부터 로봇의 위치 후보를 검색하고, 식 (4)의 적합도 (F_c)를 측정하여 위치 판단의 애매성을 줄인다.

$$F_c = \sum_{i=1}^{n-1} L(c_i, c_{i+1}) \quad (4)$$

2.5 계획 및 이동

2차원 지도에서의 경로 계획은 목표점까지의 최단 거리의 경로를 계산하는 것이다. 그러나 본 논문에서는 목표점이 단지 장애물 사이의 위치로써 표현되므로, 로봇이 목표점에 도달하기 위해서는 장애물의 정보를 획득하며 장애물을 따라 이동해야만 한다. 즉, 로봇이 목표점까지 이동하는 과정에서 로봇이 감지하는 장애물들이 로봇이 이동하는 경로가 된다. 따라서 참고문헌 [1]에서 제안한 경로계획 알고리즘과 이동 방법에 따라 경로를 선택하고 목표점까지 이동한다.

3. 모의실험

제안한 알고리즘의 타당성을 보이기 위해 컴퓨터 모의실험을 실시하고 그 결과를 보인다. 모의실험을 위해 그림 10의 로봇의 자율주행 시뮬레이터를 구현하였다. 시뮬레이터는 ①장애물이 배치되어 있는 가상의 환경, ②로봇이 감지한 주변 장애물의 카메라 화면, ③감지한 장애물과 로봇과의 관계 정보 표현, ④지도 온톨로지로부터 추론을 통해 로봇이 감지한 장애물의 후보, 그리고 ⑤로봇의 현재 위치로부터 목표점까지 이동하기 위한 경로 탐색 결과 화면으로 구성된다.

시뮬레이터의 가상 환경에 16개의 장애물을 그림 10(①)과 같이 배치하고, 로봇이 'Start' 지점에서 출발하여 'Finish' 지점(장애물 'S16')으로 자율주행 하도록 하였다. 그리고 특징이 동일한 장애물들을 'S8'의 장애물을 기준으로 대칭되도록 배치함으로써 로봇이 지도 온톨로지를 참조하더라도 현재 위치를 정확하게 판단하지 못하도록 하였다. 즉, 정확한 위치 판단을 위해 더 많은 장애물을 탐색하게 될 것이고, 그 과정에서 누적된 장애물 정보를 통해 로봇의 현재 위치를 추론할 것이다.

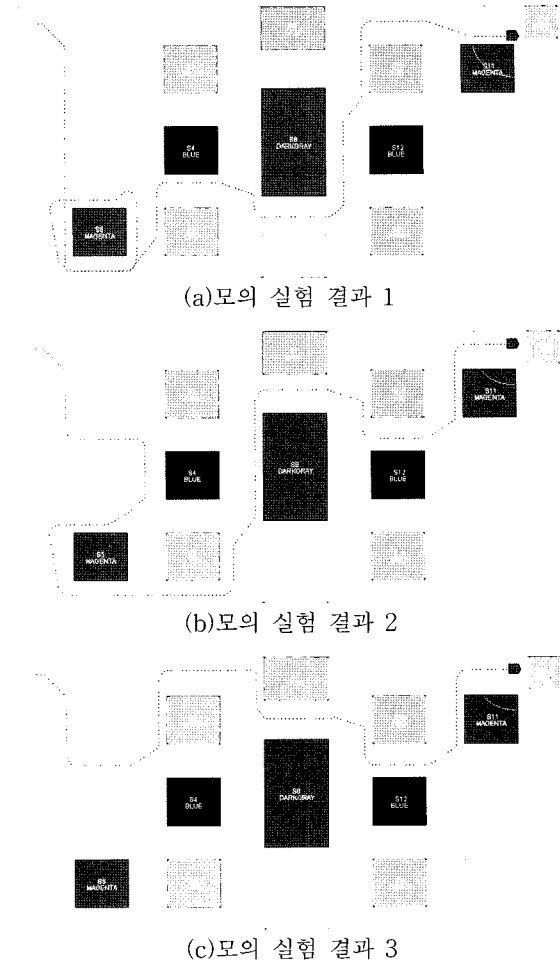


그림 11. 가상 환경에서의 모의실험 결과
Fig. 11. Simulation results on virtual environment

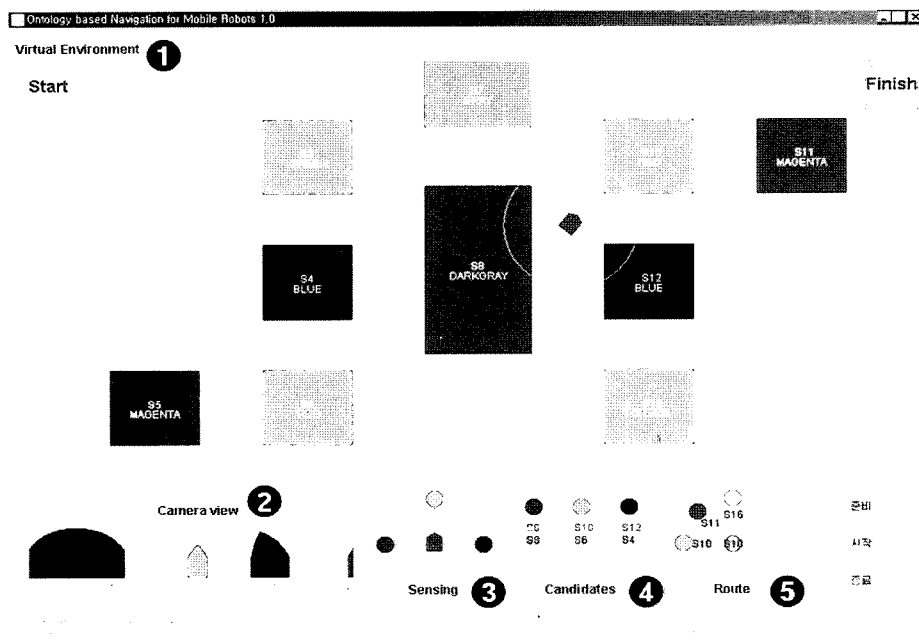


그림 10. 자율주행 로봇의 자율주행 시뮬레이터
Fig. 10. Autonomous navigation simulator for autonomous robots

그림 11은 가상 환경에서의 모의실험 결과로써 로봇이 목표점까지 이동한 3가지 형태의 궤적을 보인다. 그림 11(a)에서 로봇이 최초로 'S1'을 감지하였을 때, 지도 온톨로지로부터 'S1'과 동일한 특징의 장애물을 검색하여 3개의 장애물 후보('S1', 'S9', 'S15')를 얻었다. 이동 로봇은 3개의 장애물 후보 중 'S15'를 감지한 장애물이라 가정하고 목표점을 찾아 이동하는 형태를 보인다. 즉, 로봇은 'S15'→'S13'→'S11'→'S16'의 경로로 이동하려고 하였고, 지도 온톨로지에서의 'S15'→'S13'→'S11'와 'S1'→'S3'→'S5'의 장애물 속성이 동일하여 로봇은 잘못된 판단을 신뢰하고 이동을 계속하였다. 그러나 'S5'를 탐색하는 과정에서 목표점인 'S6'을 찾지 못하였기 때문에 잘못된 판단을 수정한다. 그리고 새로운 경로를 탐색하여 목표점까지 이동하게 된다.

그림 11(b) 또한 'S1'을 'S9'로 판단하여 'S9'→'S14'→'S13'→'S11'→'S16'의 순으로 장애물을 찾아 이동하려고 하였고, 'S9'→'S14'→'S13'→'S11'과 'S1'→'S2'→'S3'→'S5'의 속성이 동일하여 로봇은 잘못된 판단을 신뢰하고 이동을 계속하였다. 그러나 'S5' 지점에서 판단을 수정하여 새로운 경로를 탐색하여 목표점까지 이동하였다.

그림 11(c)는 잘못된 판단 없이 목표점을 바로 찾아 이동하는 것을 보인다. 수치적 좌표로 표현된 지도에서 목표점까지 최단 거리의 경로를 따라 로봇이 이동한 것과 같이, 단순히 장애물의 속성 정보와 장애물 사이의 관계 정보만으로 표현된 지도 온톨로지를 이용하였을 경우에도 로봇이 적절한 경로를 찾아 이동하는 것을 확인할 수 있었다.

4. 결 론

자율주행 로봇은 센서로부터 획득한 정보에 의존하여 주어진 목표를 수행한다. 그러나 획득한 상황 및 환경 정보에 애매성을 포함한 경우에 로봇은 주변 상황을 정확하게 인지하지 못하고 잘못된 판단을 내릴 수 있다. 또한 로봇의 행동은 예측 가능한 상황에 따라 미리 정해진 프로세스에 의해 이루어진다. 그러나 예측하지 못한 상황에 대해서는 로봇이 적절한 행동을 할 수 없는 문제가 있다.

본 논문에서는 애매성을 포함한 정보로부터 유용한 결론을 도출하기 위한 방법으로 지식과 정보를 온톨로지로 표현함으로써 로봇이 상황 변화에 유동적으로 대처할 수 있는 알고리즘을 제안하였다. 그리고 제안한 알고리즘에 기반한 모의실험을 통해 로봇이 애매한 환경 속에서도 목표점을 찾아 이동하는 것을 확인하였다.

차후 연구로는 예측하지 못한 상황에 대해서도 로봇이 적절한 행동 제어를 가능하게 하기 위한 온톨로지 제어에 관한 연구가 필요하다.

참 고 문 헌

[1] 이인근, 이동주, 이석규, 권순학, “인지에 기반한 이동 로봇의 운항계획,” *한국퍼지 및 지능시스템학회 논문지*, 제14권, 2호, pp. 171-177, 2004.
 [2] 서석태, 이인근, 권순학, “동적 환경하에서의 이동로봇을 위한 언어지도 기반 운항계획,” *한국퍼지 및 지능시스템학회 논문지*, 제14권, 4호, pp. 396-401, 2004.
 [3] A. G. Perez, M. F. Lopez, O. Corcho, *Ontological*

Engineering, Springer, 2005.

[4] SWRL: A Semantic Web Rule Language Combining OWL and RuleML, <http://www.w3.org/Submission/SWRL/>, May 2004.
 [5] SPARQL Query Language for RDF, <http://www.w3.org/TR/rdf-sparql-query/>, June 2007.
 [6] W. L. Su and S. K. Tso, “Sensor-based Fuzzy Reactive Navigation of a Mobile Robot Through Local Target Switching,” *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 29, No. 3, pp. 451-459, 1999.
 [7] P. K. Pal and A. Kar, “Mobile Robot Navigation Using a Neural Net,” *In Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 1503-1508, 1995.
 [8] 주영훈, 조상균, “유전알고리즘을 이용한 지능형 로봇의 주행 제어,” *한국퍼지 및 지능시스템학회 논문지*, 제15호, 4호, pp. 451-456, 2005.
 [9] 이현재, 최영규, “미지 환경에서 이동로봇의 주행 알고리즘,” *한국퍼지 및 지능시스템학회 논문지*, 제16권, 3호, pp. 275-284, 2006.

저 자 소개

이인근(In K. Lee)
 제17권 4호 (2007년 8월호) 참조

E-mail : iklee@ynu.ac.kr

정혜천(Hye C. Jeong)
 제17권 4호 (2007년 8월호) 참조

E-mail : doldory@kotmi.re.kr

서석태(Suk T. Seo)
 제17권 4호 (2007년 8월호) 참조

E-mail : kenneth78@ynu.ac.kr

권순학(Soon H. Kwon)
 제17권 4호 (2007년 8월호) 참조

E-mail : shkwon@yu.ac.kr