

동적 장애물 환경에서 자율운송장치의 최적 경로 계획

이 윤 배*

요 약

자율 운송 장치나 자율 이동 로봇에 대한 경로 이동 문제는 이동하고자 하는 환경을 완전히 알고 있거나, 장애물이 고정된 것으로 가정해 왔다. 따라서, 임의의 환경에 대해 부분적으로 알고 있거나 미지의 환경에서의 자율 운송 장치의 운항을 위해서는 이들 경로 기법은 직접적인 적용이 불가능하고, 확장이 어렵다. 본 논문에서는 이들 문제점을 개선하기 위해서 쿼드트리 기법을 도입하였으며, 임의의 환경에서 자율적으로 경로 계획을 할 수 있는 알고리즘을 제안하였고, 알고리즘의 타당성을 시뮬레이션을 통해 증명하였다.

An Optimal Path Planning of the Autonomous Guided Vehicle in the Environment with Dynamic Obstacles

Yun Bae Lee†

ABSTRACT

The path navigation of autonomous guided vehicle(AGV) or autonomous mobile robot(AMR) assumed that the environment was completely known and the obstacles were fixed. So that, in an environment only partly known or not known at all, the previous works were not successful since the path exploration techniques involved in the work were neither directly applicable nor extensible. In order to improve such problems, this paper was adopted the quadtree technique and proposed the algorithm for an optimal path planning autonomously in an environment and proved a validity through a simulation.

1. 서 론

임의의 환경에서 자율 운송 장치(autonomous guided vehicle : AGV)가 자율성을 갖는 시스템이 되기 위해서는 고정된 경로만을 생성하는 전략으로는 불충분하다. 또한, 과거의 경험만을 바탕으로 경로를 생성한다는 것도 한계가 있을 수밖에 없다. 특히 AGV가 이동하고자 하는 환경에 동적이거나 정적인 장애물들이 산재해 있다면, 새로운 주변 환경의 변화를 인식하여 경로를 생성해야 하는 제어 구조가 필요하다.

이를 위해 실시간 장애물 회피에 관한 여러 알고리즘이 개발되어 구현되고 있다. 일반적으로 AGV의 자율 운항은 알려진 장애물과 미지의 장애물을 가진 환경을 회복하고 실시간에 장애물 제어를 위해서 지역 경로 계획(local path planning)은 물론 알려진 장애물에 대해 경로를 생성

하는 전역 경로 계획(global path planning)을 결합하게 된다 [2].

본 논문에서, AGV는 임의의 구성 환경에 대한 맵에 의해 충돌 없는 자유 공간(collision-free space) 내에서 선택 가능한 자유 경로를 탐사하여 출발 지점에서 목표 지점까지 최적의 경로를 선택하여 운항하는 전략을 구현한다. 미지의 환경인 경우에는 환경에 대한 정보는 물론 미지의 장애물에 대한 정보가 없기 때문에, 예측 기능(expectation function), 휴리스틱 지식(heuristic knowledge)을 통해 획득한 인식 데이터에 의해 맵을 구축하여 최적 경로를 생성해야 한다[10, 11]. 이를 위해 확실성 격자 기법(certainty grid method)에 의해서 장애물을 나타내는데 이 기법은 장애물의 삽입과 운송 중에 데이터 검색이 가능하여 다른 센서와의 통합을 용이하게 한다.

본 논문에서는 확실성 격자를 생성하기 위해서 셀 크기(cell size)를 10×10 (pixel)의 정방형으로 나눈다. 각각의 cell(i, j)는 장애물이 셀 영역 내

* 본 연구는 조선대학교 '93 교비 지원에 의해서 수행됨.

† 정 회 원 : 조선대학교 전자계산학과 부교수

논문접수 : 1995년 2월 14일, 심사완료 : 1995년 4월 27일

에 존재하는 신뢰치를 나타내는 확신값(certainty value) $C(i, j)$ 를 포함하게 된다. 또한, 초음파 감지기를 이용하여 장애물의 위치에 따른 접근 가능한 거리를 계측하고, 그 결과의 분석에 따라 최적 경로를 생성하는 전략을 수행한다. 초음파 감지기는 AGV가 이동하는 동안 계속적으로 필요한 데이터를 추출한다 [2].

본 논문에서 제안하는 경로 제어 알고리즘은 AGV가 경로를 생성할 때 인간과 같이 예측의 사고 기능을 수행할 수 있도록 하였다. 다음 AGV가 실제 운항시 기계적인 오차의 누적으로 인해서 예정된 경로를 벗어날 수 있기 때문에 일정 간격마다 AGV의 이동 경로를 조사하여 오차의 누적을 예방하도록 하였다. 그리고, AGV가 예정된 경로나 미지의 환경을 주행하는 도중 경로를 이탈하였을 때 경로 회복을 할 수 있도록 하였다. 제안된 알고리즘 및 문제 해결을 위해 586급 PC와 Sun workstation을 이용하였으며, 시뮬레이션을 위한 프로그래밍 언어로 C를 사용하였다.

2. AGV의 환경

2.1 AGV의 구성

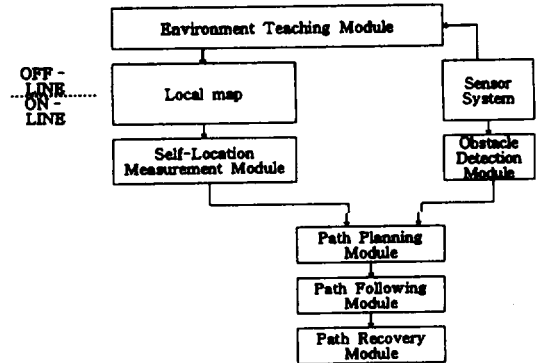
최근에 자율 이동 로봇과 AGV 시스템은 인공 지능의 한 영역으로 널리 연구가 진행되고 있다. 만약, 이들 장치가 자이로스코프(gyroscope)와 같은 내부 센서로부터 받아들이는 정보를 사용한다면, 이동 장치의 위치에 관한 오류가 누적될 수도 있기 때문에 경로를 상실하는 요인이 될 수도 있다. 더욱이, 경로 상에 예기치 않은 장애물이 출현하였을 때와 같은 환경의 변화에 유효 적절하게 대응할 수 없게 된다.

이와 같은 문제점을 개선하기 위하여 TV 카메라나 초음파 센서와 같은 외부 센서로부터 획득한 정보를 사용하는 기법이 연구되고 있다. 이들에 관한 연구는 맵에 의한 방법과 환경에 관한 지식에 의한 방법으로 다음과 같이 두 영역으로 나누어진다 [6].

(1) 환경에 관한 맵이 주어지지 않은 경우, AGV는 환경 내에서 이동 가능한 맵을 구성한다.

(2) 환경에 관한 완전하고 세부적인 맵이 주어진 경우, AGV는 맵에 의해서 경로를 추적한다.

본 논문에서는 외부센서로 초음파 센서를 이용하는 것으로 가정하며, 상기 두 가지 영역을 결합하여 AGV를 구성하는데 구성도는 (그림 1)과 같다. 또한, AGV의 동작을 OFF-LINE과 ON-LINE 등 2단계로 나누어 수행한다[1].



(그림 1) AGV의 구성도
(Fig. 1) The configuration of AGV

2.1.1 OFF-LINE 단계

OFF-LINE 단계에서 지역 맵(local map)은 AGV가 필요한 정보, 예를 들면, 장애물의 전방에 대한 데이터, 모퉁이(corners), 눈금 데이터(calibration data), 센서 처리 매개 변수, 표적 정보(target information) 등을 생성하는데 이를 위해서는 환경 교시 모듈(Environment Teaching Module ; ETM)이 필요하다.

환경 교시 모듈은 위치 조사점(location check point)에서 받아들이는 센서 데이터에 의해서 지역 맵을 생성하고, AGV가 자율적으로 이동할 때 AGV의 자기 위치(self-location)를 측정하기 위해서 사용되는 표적을 선택하는데, 지식과 정보는 이들 표적을 탐사하기 위해서 필요하다.

여기서, 환경 교시 모듈은 센서 시스템의 데이터를 이용하여 표적 탐사(target detection)와 표적 추출(target extraction)등 두 가지 작업을 수행한다.

2.1.2 ON-LINE 단계

ON-LINE 단계에서는 환경 교시 모듈에 의해

서 생성된 지역 맵과 다음 5개의 모듈이 상호 작용하여 AGV가 이동 가능한 최적 경로를 생성한다.

자기 위치 측정 모듈(Self-Location Measurement Module ; SLMM)은 AGV가 자율적으로 이동할 때 초음파 센서로부터 데이터를 입력, 지역 맵에서 보여준 표적을 탐색한다. 여기서 표적은 원, 타원형, 사각형, 직사각형, 모서리점(edge point) 등이 될 수 있는데 본 논문에서는 모서리점을 표적으로 이용하였다.

장애물 탐사 모듈(Obstacle Detection Module ; ODM)은 AGV가 이동하고자하는 임의의 환경에서 동적이거나 정적인 장애물의 존재 유무를 탐사하며 장애물이 있는 경우 장애물의 위치를 계산한다.

경로 계획 모듈(Path Planning Module ; PPM)은 출발점과 목표점 사이에서 AGV가 이동할 수 있는 경로를 계획한다. AGV의 경로 계획 도중 전방에 장애물이 발견되면, 장애물을 회피하기 위한 경로를 계획한다.

경로 추적 모듈(Path Following Module ; PFM)은 AGV가 실제로 경로를 따라 이동할 때 센서 데이터의 오독이나 기계적인 오류로 인해 예정된 경로를 벗어날 수 있기 때문에 일정 간격마다 AGV의 이동 경로를 조사하여 경로 이탈을 예방하는데 본 논문에서는 가장 근접한 모서리를 이용한다.

경로 회복 모듈(Path Recovery Module ; PRM)은 AGV가 예정된 경로를 주행하는 도중 장애물이나 기타 환경의 변화로 인해 경로를 변경시켜야 할 필요성이 발생하였을 때 이를 회복하여 예정된 경로로 이동하도록 한다.

2.2 장애물 제어 전략

2.2.1 장애물 환경 표현 모델

부분적으로 알고 있거나 완전히 미지의 환경에서 운영되는 AGV가 갖는 문제점은 센서 데이터를 가지고 환경을 수정하는 것이다. 그러나, 센서 데이터는 잡음(noise)이나 오독으로 인해 부정확성을 유발하기 쉽다. 이것은 환경 맵에 불확실성을 제공하고, 운항 업무를 보다 더 어렵게 만든

는 결과를 초래한다.

따라서, 이와 같은 문제점을 해결하고 AGV가 출발점에서 목표점까지 장애물과 충돌없이 이동 가능한 경로를 탐색하기 위해서는 효율적인 환경 표현 모델과 강력한 운항 알고리즘이 필요하다. 특히, AGV가 이동할 수 있는 환경표현 모델은 네트워크/그래프 모델(network/graph model)과 격자-기반 모델(grid-based model)로 나누어 볼 수 있다[7].

상기의 두 가지 모델이 갖는 문제점을 극복하기 위해서 [4]와 [5], [7]은 쿼드트리(quadtrees)로써 환경을 모델링하는 효율적인 기법을 제안하였다. 쿼드트리는 2진 이미지(binary image)의 효율적인 표현 기법이다. 특히, 쿼드트리는 AGV의 경로 계획을 단순화하고, 센서 데이터를 경로 계획 모델에 이용하는 것을 용이하게 하며 복잡한 환경에서도 쉽게 모델을 정할 수 있는 특징을 가지고 있다 [7].

쿼드트리는 전체 환경을 정방형 격자와 셀(square grid cell)로 분할함으로써 효율적으로 장애물 지역을 표현할 수 있다.

쿼드트리의 구성은 임의의 환경을 $N=n*n$ 의 정방형 격자(square grid)로 분할한다. 이 격자의 해상도가 초기 해상도가 되며, 각각의 초기 해상도 격자는 쿼드트리의 루트(root)가 된다. 여기서 고밀도 해상도는 환경을 $M=m*m$ 의 정방형 격자로 나눈다. 쿼드트리에서 계층 구조로 된 루트 노드(root node)는 레벨 0이 되며, 레벨은 격자를 더욱 세분화수록 증가된다. 특히, 고밀도 해상도를 가진 격자를 나타내는 노드는 레벨이 최대가 된다 [4].

충돌 문제를 배제하기 위하여, 쿼드트리는 자유 지역(free space), 장애물 지역(obstacle space), 그리고 교차 지역(intersection space)으로 각각 나타낸다.

2.3 규칙 기반 시스템

규칙 기반 시스템에 의한 AGV의 제어는 표준 프로그래밍(standard programming)과 비교할 때 다음과 같은 특징이 있다 [5, 8].

- (1) 보다 효율적인 방법을 고안하는 기능

- (2) AGV가 수행하고자 하는 기법에 관한 기능
- (3) 보다 효율적인 방법의 교시(teaching)에 관한 기능

(4) AGV가 새로운 방법을 어떻게 수행할 것인가 설명하는 기능

특히, 규칙 기반 시스템으로 표현되는 지식은 기본적인 동작이 규칙을 검색하거나, 사용하고자 하는 일련의 규칙에 의해서 설명될 수 있는데 이와 같은 특징은 AGV의 동작을 이해하고 검증하는데 필요하다.

여기서, AGV의 실질적인 이동 경로는 데이터 베이스에 저장되며 관리되는 규칙 기반 시스템의 규칙에 의해서 이루어진다.

본 논문에서 규칙 기반에 따른 규칙은 다음과 같이 3단계로 구축된다.

(1) 전역 맵 구성 규칙

- rule(1-1) IF 10×10 pixel 구역의 density > density-factor
THEN global-map [row] [column]=1로 setting
ELSE global-map [row] [column]=0으로 setting.
- rule(1-2) IF 모든 지역의 global-map 구성 완료
THEN 모서리 탐색 규칙 수행
ELSE 다른 위치의 image block을 설정하고 rule(1-1)로감

(2) 모서리 탐색 규칙

- rule(2-1) IF moving direction이 clear
THEN 예정 경로 수행
ELSE edge 탐색 후, 다음 규칙으로 감.
- rule(2-2) IF global-map [row] [column]='0'
THEN rule(2-7)로 감.
- rule(2-3) IF global-map [row-1] [column-1]='0'
and global-map [row] [column]='0'
and global-map [row+1] [column]='0'
THEN rule(2-7)로 감.
- rule(2-4) IF global-map [row] [column-1]='0'
and global-map [row] [column]='0'
and global-map [row] [column+1]='0'
THEN rule(2-7)로 감.
- rule(2-5) IF global-map [row-1] [column-1]='0'
and global-map [row] [column]='0'
and global-map [row+1] [column+1]='0'
THEN rule(2-7)로 감.
- rule(2-6) IF global-map [row-1] [column+1]='0'
and global-map [row] [column]='0'
and global-map [row+1] [column-1]='0'
THEN rule(2-7)로 감.
ELSE global-map [row] [column]=edge.
- rule(2-7) IF row, column의 위치=global-map의 끝
THEN 운항 규칙 수행
ELSE row, column을 이동후 rule(2-1)로 감.

(3) 운항 규칙

- rule(3-1) IF AGV의 현재 위치에서 goal의 가시 여부 판단
THEN goal point까지 local navigation 수행 종료.
ELSE 각 edge value 계산.
- rule(3-2) IF moving direction='clear'
THEN accelerate value를 유지하면서 이동하고
rule(3-4)로 감
ELSE rule(3-3)으로 감.
- rule(3-3) IF t 초 후에 충돌 가능성 여부 판단
THEN accelerate value를 감소하고 rule(3-2)로 감
ELSE accelerate value를 유지하며 진행하고 rule(3-2)로 감.
- rule(3-4) IF AGV의 위치=subgoal(선택된 모서리)
THEN rule(3-1)로 감
ELSE rule(3-2)로 감.

3. 최적 경로 계획

3.1 문제의 정의

AGV의 최적 경로 전략은 정적이거나 동적인 장애물이 있는 임의의 환경이나 구성 공간에서 AGV가 장애물과 충돌없이 출발 지점에서 목표 지점까지의 경로 운항이 가장 합리적으로 수행될 수 있도록 최단 경로를 선택하는 기법으로 정의할 수 있다.

본 논문에서는 이를 위해 초음파 센서를 이용하여, 초음파 센서의 문제점 즉, 오독이나 반사각이 클 때 거울 반사와 같은 문제점을 최소화하기 위해 초음파 센서 주사 거리(scanning distance)를 근거리는 20cm, 장거리는 2m 이내로 제한한다. 따라서 초음파 센서의 제한된 주사 거리내에서 가장 근접한 장애물의 모서리를 탐사하여 경로를 이동하도록 하는데 이들은 다음과 같이 정의한다.

Cs(전체 구성 공간), Qs(출발 지점), Qg(목적 지점)에 의해서, $Qs \in Cs$ 이고, $Qg \in Cs$ 일 때,

- (1) 자유 공간에 관련된 문제는 Cs를 찾거나, 이를 정의한다.
- (2) 자유 경로 탐색 문제는 $Qs \in E$, $Qg \in E$ 인 Cs의 부분 집합인 E를 찾는데 있다.

Topology 이론에서 Qs와 Qg가 분할된 공간에서 동일한 연결 요소에 속한다면, 적어도 하나의 해가 있음을 알 수 있다. 경로 계획은 다음과 같은 규칙에 의해서 수행된다.

[규칙 1]. Cs를 인식한다.

[규칙 2]. Cs를 부분집합 {C1, C2, ..., Cn}으로 분할한다.

[규칙 3]. AGV가 회전하는 교차로나 road의 상태가 변화되는 곳에서 route를 분할한다.

[규칙 4]. route는 route에 포함된 road의 형태에 따라 section의 수를 분할한다.

[규칙 5]. AGV가 이미 알고 있는 road는 분할하지 않는다.

구성 공간을 인식한다는 것은 비전형적으로 표현되는 제한된 자유 공간을 AGV가 충돌 자유 공간으로 표현함을 의미한다. [규칙 5]의 경우, AGV가 road를 이미 알고 있는지 여부에 관한 결정은 전역 맵에 기초한다. 이 맵에 나타난 장애물은 경로를 따라 AGV가 동작하는 시간과 제한 조건을 표현한다.

그러나, 단일 공간으로서 전체 구성 공간을 인식하더라도 현재의 위치나 장애물의 출현 여부에 따라 변화될 수 있기 때문에 모든 관계를 단순화하여 기술한다는 것은 쉬운 일이 아니다.

특히, AGV의 작업장 모델은 쿼드트리 기법을 적용하여 구축하는데 (그림 2)과 같다. 쿼드트리는 AGV에 부착된 센서로부터 얻은 실제 데이터

를 이용하여 효율적으로 구축할 수 있다.

쿼드트리는 전술한 바와 같이 계층적 구조를 가지며, 장애물 지역과 자유 지역을 결정한다. 쿼드트리의 각 노드는 하나의 영역을 의미하고 장애물 영역의 집합에 대해서는 부합되는 위치에 따라 white node는 자유 공간 지역을 나타내는 자유 노드를, black node는 장애물 지역을 나타내는 장애물 노드를, 그리고 gray node는 교차 지역을 나타내는 노드를 각각 의미한다[12].

3.2 맵의 구성

3.2.1. 전역 맵의 구성

전역 맵(global map)은 AGV가 최대한의 자율성을 갖고 출발지에서 목적지로 장애물과 충돌없이 이동하기 위해서 필요하다. 전역 맵은 사전에 정적인 물체나 장애물에 대한 모든 정보를 갖도록 한다. 이 정보는 장애물이나 물체의 모서리의 좌표가 된다. 만약, 동적인 물체를 전역 맵으로 표현하고자 한다면 영역에 따라 맵의 갱신 빈도는 다르겠지만, 새로운 물체의 출현은 맵의 갱신의 요인이 된다. 이런 경우, AGV의 전체 시스템의 성능 저하는 물론 자율적인 이동 경로 계획에도 문제점을 유발할 수 있다.

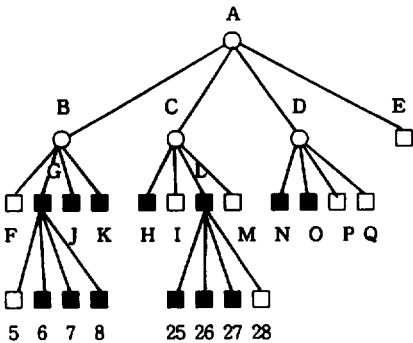
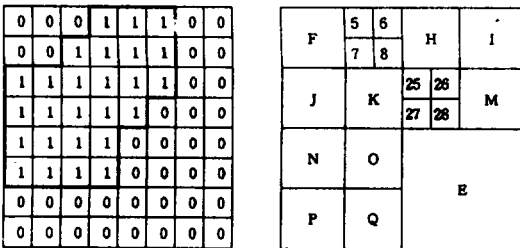
전역 맵을 구성하는 방법으로는 2차원 맵(2-dimensional map)과 3차원 맵(3-dimensional map) 등이 있다. 2차원 맵은 3차원 맵에 비하여 기억 용량면에서 많은 장점을 가지고 있으나 실질적인 물체 표현에는 제약이 있다[3]. 그러나, 공장 또는 창고와 같이 모든 저장소의 구성이 3차원적이지만 2차원의 평면을 위로 쌓아 올린 형태가 일반적이므로 2차원의 맵을 구성하는 것이 보다 효율적이다. 따라서, 본 논문에서는 2차원으로 전역 맵을 구성한다. 일단, 임의의 환경이 설정되면 행렬의 단위 크기 셀(cell)을 결정한 다음 전체 맵의 크기를 결정하는데 다음과 같다.

$$\text{Global-map} = n * n (\text{matrix})$$

$$\text{Global-map}[I, J] = 0 : \text{개방구역(자유 공간)} \dots (1)$$

$$1 : \text{폐쇄 지역(장애물 공간)}$$

전역 맵은 물체의 수에 따라 동일하게 영역을 분할한다. 예를 들면, $n * n$ 맵에서, K개의 물체가 있다면 각 물체의 영역은 $n/K * n/K$ 맵으로 구



(그림 2) 쿼드트리의 모형
(Fig. 2) The model of quadtree

성한다. 그러나, 이 방법은 간단히 영역을 분할할 수는 있으나 작업의 변화에 따라 유연하게 대처할 수 없는 문제점이 있다.

여기서, 전역 맵을 구성하는 알고리즘을 보면 다음과 같다. D(F)는 density factor를 나타낸다.

전역 맵 구성 알고리즘

```

Void global-map()
{
    int cnt=0;
    int i, j, k, l;
    for(i=1;i<=390;i=i+10)
    {
        for(j=1;j<=390;j=j+10)
        {
            for(k=i+1;k<=i+10;k++)
            {
                for(l=j+1;l<=j+10;l++)
                {
                    if(getpixel(k, l)!=0) cnt++;
                } /* l loop */
            } /* k loop*/
            if(cnt>D(F)) border [j/10][i/10]=1;
            else border [j/10][i/10]=0;
            cnt=0;
        }
    }
}
    
```

3.2.2 지역 맵의 구성

지역 맵(local map)은 AGV가 출발점에서 목표점까지 경로를 이동하는 중간에 동적인 장애물을 만났을 경우 이를 해결하기 위해서 필요한 맵으로 전역 맵과 다른 점을 요약하면 다음과 같다.

- (1) 맵의 수정이 동적이다.
- (2) 표현하는 대상이 보다 다양하다.
- (3) 휴리스틱 기법을 적용, 스스로의 탐험에 의해 새로운 경로를 생성한다.

지역 맵은 K*L행렬로 유지되며, 각 셀은 4가지의 대상을 표현하는데 다음과 같다.

$$\text{Local-MAP} = K * L \text{ (matrix)}$$

$$\text{Local-MAP}[K, L] = b_1 \ b_2 \dots\dots\dots (2)$$

$$b_1 = \begin{cases} 0; \text{free space} \\ 1; \text{obstacle space} \end{cases}$$

$$b_2 = \begin{cases} 0; \text{accessible} \\ 1; \text{not accessible} \end{cases}$$

AGV의 지역 맵은 b1, b2의 두 개의 비트로 표현되므로 (b1 b2) = (00, 01, 10, 11)의 4가지 상태 중 하나가 되는데 각각의 의미는 다음과 같다.

- (1) (b1 b2)=00은 자유 공간으로 AGV가 아무런 장애없이 통과할 수 있는 영역을 의미한다.
- (2) (b1 b2)=01은 자유 공간과 장애물 공간의 혼합 공간으로 AGV의 접근을 필요에 따라 제한하는 공간이다.
- (3) (b1 b2)=10은 AGV가 통과할 수 없으나, 접근은 가능한 공간이다.
- (4) (b1 b2) = 11은 AGV가 전혀 접근할 수 없는 영역을 의미한다.

AGV의 경로 계획은 (b1 b2)=00인 셀만이 경로로써 선택 가능하므로 AGV의 맵의 각 셀에 (b1 or b2)된 결과를 저장한 맵을 구성하며, 이와 같은 맵을 통해 노드를 결정한다.

3.3 알고리즘의 설계

3.3.1 경로 계획

AGV가 자율성을 갖기 위해서는 단순히 고정된 알고리즘 제어가 아니라 새로운 환경이나 예외적인 상황에서도 최적의 경로 계획이 가능해야 한다.

이를 위해서 AGV는 경험과 예측을 통해 적용 가능한 해를 구하는 것이 바람직하다. 휴리스틱 제어(heuristic control)는 탐험, 예측, 경험 기능이 포함된다. 휴리스틱 제어 알고리즘은 자율 시스템에서 설명되는 경로 생성 및 경로 제어에 관한 알고리즘이다. 즉, 경로 계획에서 생성되는 이동 경로는 현재 위치로부터 목표점까지의 도메인을 분할한 부분 영역(sub-space)의 경로를 먼저 결정한 다음, 각 영역의 경로를 부분적인 해(sub-solution)의 형태로 결정하여 운항 동작에

적용하는 전략을 이용한다. 도메인을 부분 영역으로 분할하는 기준은 AGV가 적용하는 환경에 따라 다르다.

3.3.2 경로 추적

AGV가 실질적으로 이동하는 경로는 센서의 오독이나 기계적인 계산상의 오차로 인해 계획된 경로를 이탈하게 된다. 본 논문에서는 계획된 경로로의 이동을 위해 전역 맵에 기억되어 있는 물체의 모서리 좌표를 조사점(check point)으로 이용한다. 즉, 계획된 경로 사이에 있는 통과 예정 경로에서 가장 근접한 모서리를 계속적으로 감시하여 오차의 누적을 예방한다.

이를 위해 본 논문에서는 초음파 센서를 사용하며, 초음파에 의해서 입력된 정보를 통해 통과 모서리(passage edge)를 추출하고, 가능한한 직선에 가깝게 경로를 설정한다.

통과 모서리 TH에 대한 AGV의 방향과 AGV와 통과 가장자리 X사이의 거리는 다음식에 의해서 계산된다.

$$TH = \arctan(\theta_x / f),$$

$$X = L \cos(TH) / \tan(K). \dots\dots\dots (3)$$

3.3.3 경로 회복

AGV는 경로 계획 단계에서 AGV가 이동할 수 있는 경로를 전역 맵과 지역 맵으로부터 획득한다. 생성된 경로에 장애 요인이 없을 경우 AGV는 그 생성된 경로대로 이동하면 되지만, 장애물이 있는 경우 새로운 부분 경로를 생성하며 이동해야 한다. 다음은 이와 같은 경로를 회복하는 알고리즘이다.

경로 회복 알고리즘

- (step 1) 현재 이동 불가능한 경로(Nx, Ny)를 결정하고 충돌 여부 검사 알고리즘을 수행. (여기서, Nx는 현재 AGV의 위치의 노드, Ny는 Nx의 예정된 경로임.)
- (step 2) Nx를 현재 테스트 노드로 결정하고 O(t)와 U(t) 크기 비교
(여기서, O(t)는 장애물의 지체 시간, U(t)는 AGV의 지체 시간임.)
 (2-1) O(t) <= U(t) 이면 (step 4) 수행
 (2-2) O(t) > U(t) 이면
 근접점 알고리즘을 수행하고 다음 단계로 감.
- (step 3) 이동 벡터 알고리즘에 의해 subgoal 결정.
- (step 4) 원래의 계획된 경로(회복) 수행.

AGV가 경로를 이동하는 도중에 동적인 장애물이 정적인 장애물로 환경이 변화되었을 때 충돌 예방을 위해서 각 세그먼트 당 1 초씩, 약 10 초 동안 장애물의 상태를 파악하여 장애물과 AGV의 충돌 여부를 검사해야 하는데 t초 후의 충돌 검사는 다음 알고리즘을 이용한다.

t 초 후의 충돌 검사 알고리즘

```

agv.prev.x, agv.prev.y : agv의 전(前) 위치
agv.curr.x, agv.curr.y : agv의 현재(現在) 위치
agv.post.x, agv.post.y : agv의 후(後) 위치
obs.prev.x, obs.prev.y : obs의 전(前) 위치
obs.curr.x, obs.curr.y : obs의 현재(現在) 위치
obs.post.x, obs.post.y : obs의 후(後) 위치
agv.delta.x, agv.delta.y : agv의 속도
obs.delta.x, obs.delta.y : obs의 속도
agv.delta.x = agv.curr.x - agv.prev.x
agv.delta.y = agv.curr.y - agv.prev.y
obs.delta.x = obs.curr.x - obs.prev.x
obs.delta.y = obs.curr.y - obs.prev.y
AGV.ACC = BRAKE / *DEFAULT ACCELATION IS BRAKE *
    
```

```

(step 1) TIME = TIME + 1
      agv.post.x = agv.curr.x + agv.delta.x
      agv.post.y = agv.curr.y + agv.delta.y
      obs.post.x = obs.curr.x + obs.delta.x
      obs.post.y = obs.curr.y + obs.delta.y

(step 2)
DISTANCE = SQRT(AGV.POST.X - OBS.POST.X)²
          + (AGV.POST.Y - OBS.POST.Y)²
IF DISTANCE < FACTOR(D)
  THEN AGV.DELTA.X = AGV.DELTA.X + AGV.ACC
       AGV.DELTA.Y = AGV.DELTA.Y + AGV.ACC
IF (COLLISION())
  THEN TIME = 0
       AGV.ACC = SPEED_UP
       GO TO (step 1)
ELSE CALL (step 2)
ELSE NEXT STEP
IF TIME > FACTOR(TIME)
  THEN RETURN /* T초 후 충돌 여부 규칙
              ELSE GO TO (step 1). 종료 */
    
```

4. 실험 및 결과 고찰

4.1 제약 조건 및 실험 범위

제한된 분석 모델의 실험을 통해 본 논문에서

제안한 알고리즘의 유효성과 타당성을 검증한다. 실험을 위해서 다음과 같은 제약 조건 및 실험 범위가 필요하다.

- (1) AGV의 경로 계획이 적용될 임의의 환경 모델을 설정한다.
- (2) 설정된 환경 모델을 토대로 AGV의 맵을 구성한다.
- (3) AGV 맵은 400×400의 크기로 40개의 sub-space를 가지며, 각 sub-space는 10×10(pixel)의 크기를 갖는다.
- (4) AGV 맵으로부터 진행될 경로의 시작점과 목표점을 결정한다.
- (5) 임의의 환경에 나타난 장애물은 정적 장애물과 동적 장애물로 구분한다.
 - (5-1) 정적 장애물의 경우 전역맵이 갖고 있는 물체의 모서리 좌표에 의해서 파악한다.
 - (5-2) 동적 장애물(또는 또다른 AGV)의 이동 속도는 AGV와 비슷한 것으로 가정하며, AGV와 장애물과의 관계는 다음과 같이 설정한다.
 - (5-2-1) 교차 지점에서 AGV 진입이 장애물보다 우선일 때는 AGV는 정상 속도로 이동한다.
 - (5-2-2) 교차 지점에서 장애물 진입이 우선일 때는 t초 후의 충돌 검사 알고리즘에 의해서 이동 경로를 결정한다.
- (6) 입력되는 실세계 모형은 AGV에 의해서 미리 학습되어 전역 맵으로 표현된다.
- (7) AGV모형은 2차원 형식으로 한다.

4.2 실험 및 결과 관찰

4.2.1 실험

본 논문에서 제안한 알고리즘의 타당성을 검증하기 위해서 C언어를 이용하여 프로그래밍을 하였으며, 실험을 위해 작성된 소프트웨어는 다음과 같이 5가지 모듈로 크게 분류할 수 있다.

- (1) 맵 구성 모듈(map construction module)은 획득된 이미지를 가지고 AGV가 이동 경로를 결정할 수 있도록 전역 맵으로 변환시키

는 루틴(routine)이다.

- (2) 모서리 탐색 모듈(edge detection module)은 전역 맵 상에 있는 일정 범위 내의 물체(또는, 장애물)의 모서리를 탐색하는 루틴이다.
- (3) 서브-골 탐색 모듈(sub-goal detection module)은 현재의 서브-골에서 목표점의 감지 여부를 판단하여 목표점이 감지되면 경로 수행을 완료하고, 그렇지 않을 경우 다른 서브-골을 탐색하는 루틴이다.
- (4) 운항 모듈(navigation module)은 초음파 센서로 물체(또는, 장애물)의 모서리를 탐색한 후 실제로 생성된 서브-골까지 경로 이동을 수행하는 루틴이다.
- (5) 장애물 회피 모듈은 맵상에서 AGV와 장애물과 t초 후에 충돌 여부를 판단하여 AGV의 진행 속도를 가속, 감속 또는 현재 속도 유지 등으로 구분하여 장애물을 회피하는 임무를 수행하는 루틴이다.

이와 같은 대표적인 5가지 모듈을 기본으로 하여 다음과 같은 실험을 통해 제안된 알고리즘의 타당성과 유효성을 검증하기로 하겠다.

[실험 1] 정적인 장애물이 있을 때의 경로 이동 : 주로 전역 맵을 참조하게 되는데, AGV는 장애물과의 가장 근접한 모서리를 탐색, 직선으로 연결, 각의 값이 가장 작은 근접점을 경로로 선택하여 이동하게 되는데 실험 결과는 (그림 3)과 같다.

[실험 2] 동적인 장애물이 있을 때의 경로 이동 : AGV가 이동하는 도중에 동적 장애물을 만났을 때 모서리 좌표를 센서 시스템에 의해서 추론하며, 이동 벡터 계산 알고리즘과 t초 후의 충돌 검사 알고리즘에 의해서 거리 및 시간을 계산하고 경로 계획 알고리즘을 수행하는데 실험 결과는 다음과 같다.

[실험 2-1] 동적 장애물이 AGV 좌우측에서 접근해 올 때 : (그림 4)

[실험 2-2] AGV 경로가 변경되고 동적 장애물이 좌측에서 우측으로 이동할 때 : (그림 5)

[실험 2-3] AGV 경로가 변경되고 동적 장애물

이 우측에서 좌측으로 접근해 올 때 : (그림6)

4.2.2 결과 고찰

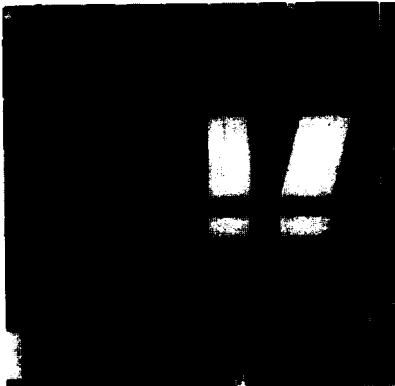
본 논문에서 제안한 AGV의 최적 경로 제어 시스템은 AGV가 맵에 의해서 경로를 생성, 출발점에서 목표점까지 이동할 때 인간과 같이 예측의 사고 기능을 갖도록 하였다. 또한, AGV가 실제로 경로 이동시 센서의 오독이나 기계적인 오차의 누적으로 인해 예정된 경로를 벗어날 수 있기 때문에 일정 간격마다 가장 근접한 모서리를 조사점(check point)으로 설정하여 검사함으로써 오차의 누적을 최소화하였다.

그리고, 정적 장애물의 위치 좌표는 전역 맵을 참조하도록 하였으며, 동적 장애물은 지역 맵을 참조하도록 하였다.

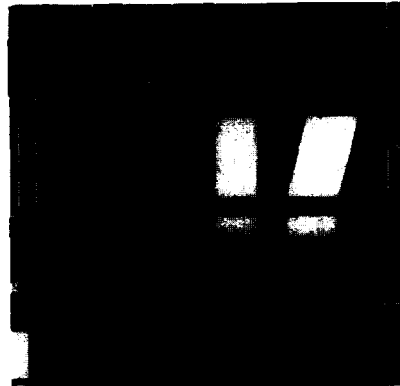
쿼드트리 접근법을 도입, AGV가 경로 이동시 임의의 주어진 환경을 자유 공간, 장애물 공간 등으로 분할함으로써 AGV와 장애물과의 충돌을 사전에 예방하도록 하였으며, AGV의 속도는 1u/sec로 가정하였고, 동적 장애물의 속도도 이와 같이 가정하여 실험하였다. 여기서, 1u는 1unit로서 1u/sec는 초당 10×10(pixel) 이동 거리를 의미한다.

특히, AGV가 경로 이동 중에 동적 장애물을 만났을 때 경로 계획을 다음과 같이 수행하도록 하였다.

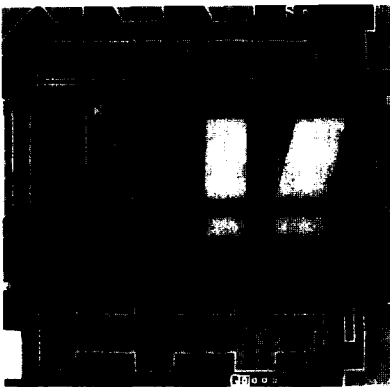
- (1) AGV와 동적 장애물의 교차 지점에서 AGV의 진입이 장애물보다 우선일 경우 AGV는 원래 속도로 이동하도록 하였다.
- (2) AGV와 동적 장애물의 교차 지점에서 장애물 진입이 AGV보다 우선일 경우 AGV는



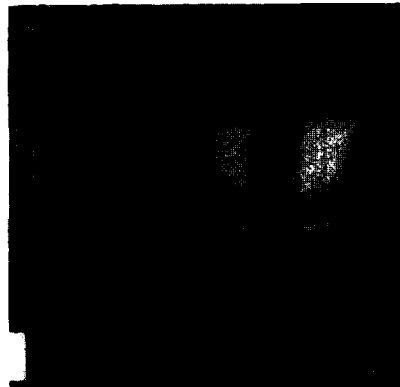
(그림 3) 실험 1의 결과
(Fig. 3) The result of experiment [1]



(그림 5) 실험 2-2의 결과
(Fig. 5) The result of experiment [2-2]



(그림 4) 실험 2-1의 결과
(Fig. 4) The result of experiment [2-1]



(그림 6) 실험 2-3의 결과
(Fig. 6) The result of experiment [2-3]

예정 경로로 탐사된 모서리 점을 찾아 이동하며, 장애물의 위치에 따라 좌나 우로 장애물을 회피하여 경로를 회복하도록 하였다. 그리고, 동적 장애물이 이동 중에 AGV 전방에 멈추어 섰을 때 t 초 후의 충돌 검사 알고리즘에 의해서 일정 시간(본 논문에서는 약 10초) 동안 대기 후 정적 장애물로 간주하여 근접점 알고리즘을 이용하여 경로를 찾고 경로 회복 알고리즘을 이용하여 원래의 경로를 회복하도록 하였다.

- (3) 교차 지점에서 AGV와 동적 장애물이 동시에 진입할 경우에는 동적 장애물과 반대 방향에 있는 모서리를 근접점으로 이용하여 장애물과 비슷한 속도로 이동하도록 하였다.

상기와 같은 방법으로 실험하였을 때 본 논문에서 제안한 AGV로부터 얻을 수 있는 기대 효과를 요약하면 다음과 같다.

첫째, 임의의 환경을 자유 공간, 장애물 공간 등으로 나누어 장애물에 대한 상태의 분석이 가능하므로 동적인 장애물과 마주치는 경우에도 예정된 경로를 유연하게 수행할 수 있다.

둘째, 정적인 장애물의 경우 지식베이스에 저장된 지식과 전역 맵에 의해서 최적 경로를 선택하여 이동 가능하므로 이동 시간을 단축할 수 있다.

셋째, 초음파 센서 시스템을 채용, 일정 시간마다 근접점(모서리)를 검사함으로써 계획된 경로의 이탈 오차를 극소화할 수 있다.

넷째, 쿼드트리 기법에 의해서 이동 공간을 분할, 다음에 이동할 구간에서의 장애물과 충돌 가능성을 사전에 검사함으로써 실시간 제어를 할 수 있다.

다섯째, 예정된 경로를 반복 주행하는 단순한 AGV를 탈피하여 임의의 환경이나 작업장의 확장 및 변경이 용이하여 비용 절감 효과를 기대할 수 있다.

5. 결 론

경로 계획은 AGV의 실현에 있어서 필수 불가

결한 문제의 하나로 고려되고 있다. 따라서, 본 연구에서는 어느 정도 구조화된 작업장이나 어수선하지 않는 환경에서 장애물을 회피, 최적 경로 계획을 수행하는 AGV를 제안하였다.

AGV의 구성은 OFF-LINE 단계와 ON-LINE 단계로 구분하였고, 특히, ON-LINE 단계는, 자기 위치 측정 모듈, 장애물 탐사 모듈, 경로 계획 모듈, 경로 추적 모듈, 경로 회복 모듈 등을 독립된 모듈로 구성, 기존의 자율 운송 장치의 계층적 제어 구조에서 상위 레벨이 하위 레벨의 제어를 항상 감독해야 하는 부담을 최소화하고 각 모듈 별로 독립성을 갖도록 함으로써 성능의 향상을 기하였다.

AGV가 계층적 제어를 수행함으로써 AGV는 자신의 작업 환경에 관해 선 지식을 갖게 되며, 따라서, AGV가 예측한 경로와 일치되지 않는 상황에서도 신속하게 경로 계획을 할 수 있도록 하였으며, 동적 장애물은 지역 맵을 통해서 최적 경로를 수행할 수 있도록 하였다. 또한, AGV의 운항 경로를 몇 개의 영역으로 분할하여 다음에 이동하게 될 구간에서 충돌 가능성을 검사하도록 하였는데 여기서는 쿼드트리 기법을 이용하였다. 아울러, AGV가 이동 가능한 경로 선택을 위해서 과거의 경험을 저장한 지식베이스와 가중치를 적용하여 인간의 휴리스틱 기능과 같은 능력을 최대한 발휘할 수 있도록 하였다. 그러나, 휴리스틱 기능은 환경의 변화가 심하고 비결정적인 경우 또는, 환경 전체를 사전에 알수 없는 경우에는 탐험 기법을 이용해야 한다.

이와 같은 문제점의 보완과 함께 향후 연구 과제를 요약하면 다음과 같다.

첫째, 자기 위치 측정 모듈(SLMM)에서 처리 시간을 단축할 수 있어야 하고, 다각형은 물론 원형이나 타원형과 같은 특징을 신속히 검출할 수 있는 중간 단계의 시각 처리기(vision processor)가 고려되어야 한다.

둘째, 인간과 AGV와의 인터페이스 도구로서 음성 인식을 이용한 명령 전달과 AGV의 출력을 음성 합성으로 이용할 수 있는 verbal interface에 관한 연구가 있어야 한다.

셋째, 임의의 환경에서 장애물 출현을 자동으로 전달할 수 있고 진단 결과를 자동으로 처리할

수 있는 자동 규칙 생성에 관한 연구가 필요하다.

네째, 비전 시스템과 초음파 센서 또는 적외선 센서를 결합하여 보다 복잡한 환경이나 다수의 장애물이 이동하는 환경에서도 정확히 최적 경로를 계획할 수 있는 알고리즘의 개발이 필요하다.

다섯째, 퍼지 추론(fuzzy reasoning)을 이용하여 AGV의 운항과 장애물 출현에 관한 지식을 획득할 수 있고, 학습 기능을 적용할 수 있는 연구가 필요하다.

참 고 문 헌

[1] M.D.Adams, Huosheng Hu and P.J.Probert, "Towards A Real-Time Architecture for Obstacle Avoidnce and Path plannin g in Mobile Robots", In IEEE J.Robotics and Automation, pp. 584-589, 1990.

[2] J.Borenstein and Yorem Koren, "Real-Ti me Obstacle Avoidance for Fast Mobile Robot", IEEE Trans, of System, Vol. 19, No. 5, pp. 1179-1187, 1989.

[3] J. Borenstein and Y.Koren, "Real-Time Obstacle Avoidance for Fast Mobile Rob ots in Cluttered Environments", IEEE In formation Conference on Robotics and A utomation, pp. 572-576, 1991.

[4] R.H.T Chan and P.K.S Tam, "Robot Na vigation in Unknown Terrains via Multi-resolution Grid-maps", IECON'91, pp. 1138-1143, 1991.

[5] Yun-Hui Liu and Suguru Arimoto, "A Fl exible Algorithm for Planning Local Sho rtest Path of Mobile Robots Based on Reachability Graph", IEEE Interational Workshop on Intelligent Robots and Systems, IROS, pp. 749-757, 1990.

[6] Kazumori Onoguchi and Mutsumi Watan able and Yasukazu Okamoto, "A Visual Navigation Using a Multi-Information Local

Map", In Proc. of USA-JAPAN symposium on Flexible Automation, pp. 767-774, 1992.

[7] Alexander Zelinsky, "A Mobile Robot Explor ation Algorithm", IEEE Trans. on Robotics and Automation, Vol. 8, No. 6, pp. 707-717, 1992.

[8] 이윤배, 전문가시스템, 흥릉과학, pp. 41-69, 1992.

[9] 이윤배, "자율 운송 장치의 휴리스틱 제어 전략", 한국정보과학회(호남, 제주) 학술발 표논문집, 13권 1호, pp. 57-60, 1991.

[10] 이윤배, 나영남, "휴리스틱 제어를 적용한 자율 운송 장치의 경로계획", 한국정보과학 회 학술발표논문집, 제 19권 1호, pp. 181-184, 1992.

[11] 이윤배, 김상삼, "자율 운송 시스템의 최적 관리 계획", 한국정보과학회(호남, 제주) 학 술발표논문집, 제4권 1호, pp. 347-356, 1992.

[12] 이윤배, 이옥빈, "자유 공간에서 움직이는 장애물 회피를 위한 자율 운송 장치의 제어 시스템", 한국정보과학회 학술발표논문집, 제 20권 1호, pp. 233-236, 1993.

이 윤 배

1980년 광운대학교 전자계산학과 졸업(이학사)
 1983년 광운대학교 대학원 전자계산학과 졸업(이학 석사)
 1993년 숭실대학교 대학원 전자계산학과(공학 박사)
 1979년~81년 태평양화학(주) 전산실 근무
 1981년~85년 서울 영락상업고등학교 교무주임
 1985년~88년 군산전문대학 전산소장
 1988년~현재 조선대학교 전자계산학과 부교수
 관심분야 : 인공지능, 전문가 시스템, 로보틱스, 멀티 미디어 등.