

작업환경 모델 기반 AGV의 최단 경로 탐색 알고리즘

Shortest Path Searching Algorithm for AGV Based on Working Environmental Model

주영훈 · 김종선

Young Hoon Joo and Jong Seon Kim

군산대학교 전자정보공학부

요약

본 논문에서는 AGVS(Automated Guided Vehicle System)가 여러 복잡한 작업 환경 또는 작업 환경 변경 시 좀 더 유연하게 운용될 수 있도록 작업환경 내에서 AGVS에 필요한 작업공간요소를 분류하고 이들을 모델링하는 방법을 제안한다. 또한, 그래프 탐색 방법인 A* 알고리즘을 이용하여 AGV의 최단 경로 탐색 알고리즘을 본 논문의 작업환경 요소로서 재표현한다. 생성된 최단 경로와 본 논문에서 가정한 AGV의 속도 테이블을 이용하여 운행 중인 AGV의 경로 점유 시간 알고리즘을 제안한다. 마지막으로 간단한 시뮬레이션을 통하여 제안한 방법의 적용 가능성을 증명한다.

Abstract

This paper proposes the effective method for classifying the working spaces and modelling the environments, when complex working environments of AGVS(Automated Guided Vehicle System) are changed. And, we propose the shortest path searching algorithm using the A* algorithm of graph search method. Also, we propose the methods for finding each AGV's travel time of shortest path. Finally, a simple example is included for visualizing the feasibility of the proposed methods.

Key Words : Automated Guide Vehicle(AGV) , A* algorithm, travel time, heuristic search, shortest path

1. 서론

물류자동화의 기술에서 빠질 수 없는 것이 바로 AGV(Automated Guided Vehicle)이다. AGV는 작업 공간 내에서 이동시켜야 할 물류들을 자동으로 이동 시켜주는 이동로봇으로써, 1953년부터 사용되어지기 시작했다. 기술의 발전에 힘입은 FA(Factory Automation)의 가속화는 유연 생산 시스템(FMS : Flexible Manufacturing System) 및 유연 조립 시스템 (FAS: Flexible Assembly System)을 더욱 추구하게 되었으며 AGV의 운용은 제품의 단순 물류작업에서 공정 간의 제품의 이송까지 그 범위가 확대 되었다[1, 2]. 특히, 작업 공정 간의 물류이동은 생산되는 제품의 공정 수에 따라 더욱 복잡해질 수 있다. 다수의 AGV를 사용하는 대규모 운송시스템에서는 복잡 다양한 물류이동을 효과적으로 처리하기 위한 알고리즘 개발을 요구하고 있으며 AGV 간의 충돌 고려, 물류운송 최단 경로 생성 및 경로 유도가 알고리즘의 주요 핵심이라 할 수 있겠다[3]. 따라서 본 논문 역시 대규모 운송 시스템에서 다수의 AGV를 효과적으로 운용하기 위한 알고리즘 개발에 그 초점을 두고 있다.

기존의 논문에서 대부분의 경로 기법은 다수의 AGV들에 대한 경로를 충돌을 고려하지 않은 최단 경로를 생성하고 AGV를 유도하는 도중 발생하는 교착 및 충돌의 상태를 영역 제어(Zone-Control) 방식으로 관리한다[4]. 영역 제어 방식은 작업 공간에서 서로 인접한 경로 및 교착 예상 구역으

로 나눈 다음 이들에 대한 점유권을 오직 1개의 AGV로 제한함으로써 경로상의 충돌 문제를 해결한다. 김상휘, 황학은 AGV의 효율적인 운영을 위해 자치제어구조에서 작업할당 AGV를 선택함에 있어 입찰 함수의 개념을 도입하였고, 격자 탐색을 이용한 급송 함수를 사용하여 AGV 급송규칙에 적용 하였다[7]. 입찰 함수는 여러 개의 작업요청이 있을 경우 입/출고대의 작업물 정보, 이동거리를 수치적으로 환산하여 작업을 선택하는 것이다. 그러나, 상기와 같은 경로 선정의 기법은 충돌 및 교착을 예상하지 않은 즉, 장애물에 의해 지연되는 시간을 고려하지 않고 있다. 이는 작업물의 총 운송시간을 증가시키게 되고 운송시스템의 효율을 저하시킨다. 장애물에 의해 발생하는 지연시간을 최소화하고자 Moorthy[6]는 장애물의 위치 및 교착지점을 예측하여 영역 제어 방식에서 발생하는 지연시간을 최소화 하였으며, Rajotia[7]는 강제적 경로 배정계획을 통해 예지 및 노드가 갖는 Time-Window에서 AGV의 교착을 예측하는 알고리즘을 제안하였다. 또한, 최이 등[13]은 컨테이너 터미널에서 AGV의 충돌 및 데드락, 라이브락을 방지하기 위해 AGV의 점유영역 예약 테이블(Occupancy Area Reservation table; OAR table)을 이용하는 방법과 AGV의 가감속을 고려하여 AGV의 주행시간을 추정하는 방법을 제안함으로써 이를 해결하였다.

앞서 언급한 바와 같이 장애물에 의해 발생하는 운송시간 지연이 운송시스템의 효율을 저하시킴을 알 수 있다. 따라서 장애물에 대한 효과적인 처리가 운송시스템의 효율을 높이는 최선의 방법 중 하나라고 해도 과언이 아닐 것이다. 이러한 측면에서 본 연구는 장애물을 고려한 최적의 경로 생성 및

접수일자 : 2007년 5월 19일
완료일자 : 2007년 10월 5일

효율적인 교차제어를 최종 목표로 하고 있다. 하지만 그전에 선행되어야 할 과제로써 주어진 운송시스템의 작업 환경을 모델링하고, AGV와 작업장 간의 최단 경로(Shortest path)를 생성하는 알고리즘을 구현해야 한다. 그런 뒤 최단 경로를 바탕으로 충돌 예측을 위해 시간에 따른 각 AGV의 경로별 소요시간을 산출해내는 것을 본 논문의 주된 목적으로 한다.

작업 환경의 모델링에 있어서는 AGVS가 필요한 기능으로써 작업환경을 분류하고 모델링 한다. 또한, AGV와 작업장간의 최단 경로 생성을 위한 알고리즘은 그래프 탐색 기법인 BFS(Best first search) 및 Dijkstra, A*알고리즘 중 시스템의 자원을 가장 적게 사용하면서 최적의 최단 경로를 생성하는 A* 알고리즘을 이용하여 최단 경로 탐색 알고리즘[8, 9, 10]을 구현하고 이를 본 논문에서 언급한 작업 환경 모델로서 재 표현 한다. 이렇게 생성된 최단 경로에서 AGV의 속도 및 가속도를 정의 하고 AGV의 시간에 따른 경로별 점유시간 및 경로 진입 방향, 경로 접합 형태를 산출함으로써 충돌 및 교차를 예측할 수 있음을 보인다.

2. AGVS의 작업환경 모델

먼저 작업환경의 요소를 AGVS가 지녀야할 기능 및 데이터 모델로서 살펴본다. Pooya 등[14]은 AGVS를 제어하기 위해 Multi-Agent System(MAS)를 제안하고 이들을 Multi Cell Agent, Scheduler Agent, Material Manager Agent, Material, Manager Agent로 분류하였다. Pooya 등이 제안한 작업환경의 데이터 모델은 본 논문에서 요구하는 데이터 모델과 상충한다. 본 논문에서 제안하고자 하는 모델은 Pooya 등이 분류한 작업 환경 모델로 분류하되 AGVS가 지녀야 하는 기능을 고려한 모델이다. AGVS가 지녀야할 기능은 다음과 같이 3가지로 요약할 수 있다.

1. AGVS가 적용된 작업 환경 내에서 작업장의 요청에 따라 AGV의 최단 경로를 생성한다.
2. 각 AGV를 이동경로 상에서의 충돌이 발생하기 이전에 이를 예측하고 충돌을 회피하도록 유도한다.
3. 모든 AGV와 통신하며 정보를 송수신하여 AGV의 상태를 관리 및 감독한다.

이와 같은 AGV의 기능적 측면과 AGVS의 데이터 모델 측면을 고려한 AGV의 작업 환경 요소는 경로, 작업장, 통제구역, AGV, 운송물류로 분류할 수 있다. 최단 경로를 생성하기 위해서는 작업을 요청한 작업장의 위치정보와 경로의 정보가 필요하며 운행 중인 AGV 상호간의 충돌을 방지하기 위해서도 경로의 정보가 필요하며 경로가 서로 교차하는 지점 즉, 통제구역의 정보역시 필요하다. 또한 AGV는 실시간으로 운행하며 관리할 수 있는 무선통신 프로토콜 및 관리시스템이 요구 되나, AGV의 관리는 작업환경의 요소라기보다는 작업 환경 내에서 통제되어야 할 대상이기 때문에 환경요소의 분류에서 제외한다. 운송 물류는 본 논문의 최단 경로 탐색 및 소요시간 산출 알고리즘에서 시뮬레이션시의 고정된 작업 위치 데이터만을 사용하므로 작업 환경 모델에서 제외한다.

분류된 작업환경의 요소는 AGVS의 환경 데이터로 만들기 위해 각각의 고유 ID, 작업공간상에서의 위치, 거리, 명령 전환점(check point) 등의 정보를 추가로 지녀야 한다. 그림 1은 AGVS의 작업환경의 요소들을 도시한 그림이며, 분류된 작업환경의 요소는 표 1과 같이 정보를 포함하는 모델로서 나타내었다.

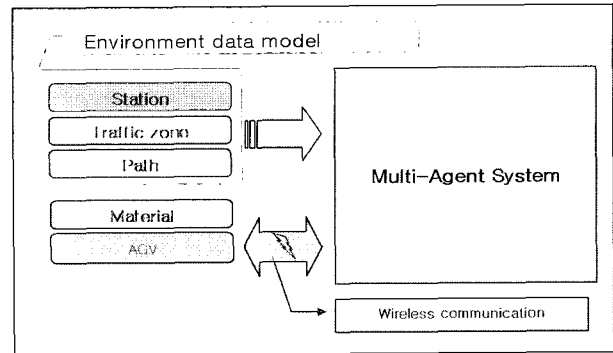


그림 1. AGVS의 작업환경 요소

Fig. 1. Elements for working environments of AGVS

표 1. 작업 환경 요소의 모델

Table 1. Model of working environmental elements.

Classification	ID	Position(pos)	Check Point (CP)	Length
Path P_i	P_{i1}	$P_{i2} = \{type, pos1, pos2\}$	$P_{i3} = \{CP_1, CP_2\}$	P_{i4}
Working station W_i	W_{i1}	$W_{i2} = \{type, pos\}$	$W_{i3} = \{CP_1\}$	W_{i4}
Traffic zone TZ_i	TZ_{i1}	$TZ_{i2} = \{pos\}$		TZ_{i3}

(단, i 는 입력된 모델의 수)

입력된 모든 작업환경요소에 대한 표현은 아래와 같이 각 요소가 지닌 특성들의 집합들로서 표시된다.

따라서 경로입력에 대한 정보 P_i 는 표 1의 내부 정보를 식 (1)과 같이 표현된다.

$$P_i = \sum_{\alpha=1}^4 P_{i\alpha} \quad (1)$$

(단, α 는 모델의 내부요소, i 는 입력된 모델의 수)

작업장(Working Station) 및 통제구역(Traffic zone) 역시 식(1)과 같은 표현으로 나타내면 식 (2), (3)과 같다.

$$W_i = \sum_{\alpha=1}^4 W_{i\alpha} \quad (2)$$

$$TZ_i = \sum_{\alpha=1}^3 TZ_{i\alpha} \quad (3)$$

3. 모델을 통해서 본 최단 경로와 소요 시간

3.1 최단 경로 (Shortest Path)

본 논문에서는 최단 경로 생성을 위한 기법으로써 A* 알고리즘을 이용한다[8-10]. A* 알고리즘은 평가함수를 사용하기 때문에 생성된 최단 경로가 항상 과소평가하도록 경로를 생성한다. 따라서 A* 알고리즘의 최적성은 조건에 따라 다르지만, 보통 다음과 같은 * 허용성의 가정아래 최적의 해를 가지는 것으로 본다.

* 허용성의 가정 : $\hat{h}(n)$ 은 실제 최단 거리 $h(n)$ 과 $\hat{h}(n) \leq h(n)$ 의 조건을 만족

A* 알고리즘은 임의의 노드 n에 대해 식 (4)와 같은 평가 함수를 갖는다.

$$f(n) = g(n) + \hat{h}(n) \quad (4)$$

여기서, $f(n)$: 평가함수
 $g(n)$: 시작 노드에서 n노드까지의 최단거리
 $\hat{h}(n)$: n노드에서 목표 노드까지의 추정치

본 논문에서는 목표노드 상에서 작업을 요청한 작업장의 위치를 w 라 하고, n번째 노드에서 목표 노드까지의 추정치 $\hat{h}(n)$ 를 $\hat{h}(n) = \sqrt{(w-n)^2}$ 로서 정의한다. 이는 탐색중인 현재 노드 n에서 평가함수 $f(n)$ 이 갖는 추정치이다. 이를 작업 환경 모델로써 표현하면, $\hat{h}(n) = \sqrt{(W_{\delta 2} - TZ_{n2})^2}$ 로서 표현된다.

여기서, δ 는 입력된 i개의 작업장 중 하나이며, $W_{\delta 2}$ 는 작업장의 위치, TZ_{n2} 는 노드 n의 위치이다.

```

open_node(n_count)=start_V;           %노드 탐색 시작
while open_node(n_count) ~= target_V %목표 노드까지 수행.
    count_j=1;
    for count_i=1:98,                 % 인접 노드 탐색.
        if g_node(count_i,1) == open_node(n_count)
            child_node(count_j)=g_node(count_i,2);
            count_j=count_j+1;
        end
    end
    n_count = n_count + 1;
    weight_vertex = [100000 100000 100000 100000]; % 평가함수 공간을-
    count_j = count_j - 1;                % ∞으로 초기화.
    for count_w=1:count_j,             % 평가 함수 h'(n) 계산
        weight_vertex(count_w) = sqrt((P_Position(child_node(count_w),1) -
        P_Position(target_V,1)).^2 + (P_Position(child_node(count_w),2) -
        P_Position(target_V,2)).^2);
    end
    for count_bs1=1:count_j,          % 평가 함수에 따른 노드 정렬.
        flag =0;
        for count_bs2=1:count_j-count_bs1,
            if weight_vertex(count_bs2) >= weight_vertex(count_bs2+1)
                temp = weight_vertex(count_bs2);
                temp2 = child_node(count_bs2);
                weight_vertex(count_bs2) = weight_vertex(count_bs2+1);
                child_node(count_bs2) = child_node(count_bs2+1);
                weight_vertex(count_bs2+1) = temp;
                child_node(count_bs2+1) = temp2;
            end
        end
        flag = 1;
    end
    end
    end
    child_node;
    open_node(n_count)=child_node(1); % 정렬된 노드 첫 번째를 다음 탐색
    open_node;                       % 시 최소 거리의 노드로 선택.
end
    
```

그림 2 최단 경로 탐색 알고리즘
 Fig. 2. Shortest path search algorithm

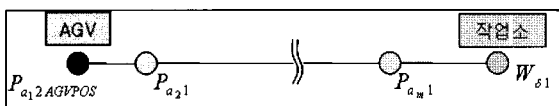


그림 3. SP_α 의 경로
 Fig. 3. Path of SP_α

또한, $W_{\delta 2}$ 는 노드에 있지 않고 경로 상에 존재 하며, A* 알고리즘은 W_δ 가 속해있는 노드 즉, 통제구역(TZ_δ)을 찾는다. 따라서 $\hat{h}(n)$ 은 $W_{\delta 4}$ (경로의 양쪽 끝에서 떨어진 거리, 표 1 참고)만큼의 여유를 갖기 때문에 허용성의 가정을 항상 만족한다. 생성된 최단 경로는 다음과 같이 식 (5)와 같이 표현된다. 그림 2는 본 논문에서 제안한 평가함수의 선정 방법을 A* 알고리즘을 적용, 최단 경로 탐색 알고리즘을 구현한 것이며 그림 3은 최단 경로 α (Shortest Path : SP_α)가 갖는 경로를 도식적으로 나타낸 것이다. 또한, 최단 경로의 총거리 D 는 경로집합 요소 표현으로 식 (6)과 같이 표현 가능하며 그림 4과 같이 나타내었다.

$$SP_\alpha = \{P_{\alpha 1,2AGVPOS}, P_{\alpha 2,1}, P_{\alpha 3,1}, \dots, P_{\alpha n,1}, W_{\delta 1}\} \quad (5)$$

여기서, $W_{\delta 1}$: 경로 $P_{\alpha n}$ 상의 목표 작업장 위치.
 $P_{\alpha 1,2AGVPOS}$: 경로 $P_{\alpha 1}$ 상의 AGV의 위치.

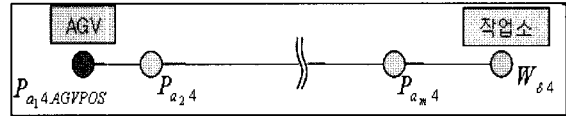


그림 4. SP_α 의 거리
 Fig. 4. Length of SP_α

$$D = W_{\delta 4} + \sum_{n=2}^m SP_{\alpha n,4} + P_{\alpha 1,4AGVPOS} \quad (6)$$

여기서, $W_{\delta 4}$: 경로 $P_{\alpha n}$ 에서부터 작업장까지의 거리.
 $P_{\alpha 1,4AGVPOS}$: AGV가 경로 $P_{\alpha 1}$ 으로부터 떨어진 거리.

3.2 최단 경로 소비 시간

선택된 최단 경로에 AGV의 이동속도와 이동방향을 찾은 뒤 이를 시간에 대한 함수로 표현하면 AGV의 시간에 따른 작업 공간상의 위치를 추적할 수 있다. 따라서 시간을 가속시킨다면 AGV가 서로 같은 경로에 서로 반대방향으로 진입하거나(정면충돌), 통제구역을 동시에 점유하여 발생하는 AGV의 충돌을 예측할 수 있다. 주어진 최단 경로의 집합에서 시간에 대한 AGV의 위치를 나타내기 위해서는 먼저 AGV의 속도와 가속도 및 통제구역의 거리(AGV의 방향 전환을 위한 감속구간)를 정의할 필요가 있다. 최이 등[13]은 가속도를 고려한 AGV의 주행경로를 연구함에 있어 AGV의 속도를 가정한다 있다. 이를 바탕으로 AGV의 속도 및 가속도의 변화를 그림 5에 나타내었으며 표 2에 정의하였다.

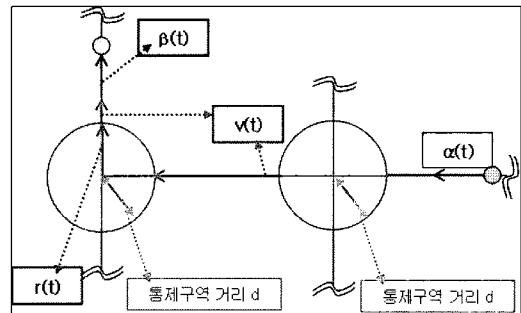


그림 5. AGV의 경로에 따른 속도 및 가속도
 Fig. 5. Velocity and acceleration by AGV's path

표 2. AGV의 속도 및 가속도의 정의
Table 2. Definition of AGV's velocity and acceleration

$v(t)$	AGV가 직선 구간을 이동할 때의 평균 속도
$\alpha(t)$	정지 후 출발 시 평균가속도
$\beta(t)$	운행 중 정지시의 평균 가속도
$\gamma(t)$	방향 전환 시 감속 운전 평균속도
d	통제 구역의 거리

3.3 최단 경로의 방향(SP_{c_i}) 및 방향 전환(n_i) 데이터

AGV의 경로 소요시간을 산출하기 이전에 먼저 AGV의 경로별 속도 변화를 살펴본다. AGV 표 2에서 정의하고 그림 5에서 보였듯이 경로간의 접합형태에 따라 AGV는 $\gamma(t)$ 로 주행할 것인지 $v(t)$ 로 주행할 것인지를 알아야 한다. 따라서 방향 전환 여부를 나타내는 데이터 및 경로 상 실제 AGV의 방향을 갖는 데이터를 나타낼 필요가 있다.

- 최단 경로 방향 SP_{c_i} :

$$SP_{c_i} \in \{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$$

여기서, θ 는 경로 방향 (Degree), i 는 최단 경로의 경로 집합 수를 나타낸다.

- 방향 전환 n_j :

$$n_j \in \{1, 0\}$$

여기서, 0은 이웃하는 경로 방향이 같을 때, 1은 이웃하는 경로 방향이 다를 때, j 는 $j=i-1$ 을 나타낸다.

그림 6은 방향 및 방향 전환 데이터를 나타낸 것으로써 임의로 정한 최단 경로 k 를 간략히 나타낸 것이다. 최단 경로 k 의 경로 개수(m)는 6개이고 그림 6과 같은 방향을 갖는 최단 경로 k 의 SP_{c_i} 및 n_j 는 다음과 같다.

$$SP_{c_i} = \{0, 90, 0, 0, 90, 180\}, n_j = \{1, 1, 0, 1, 1\}$$

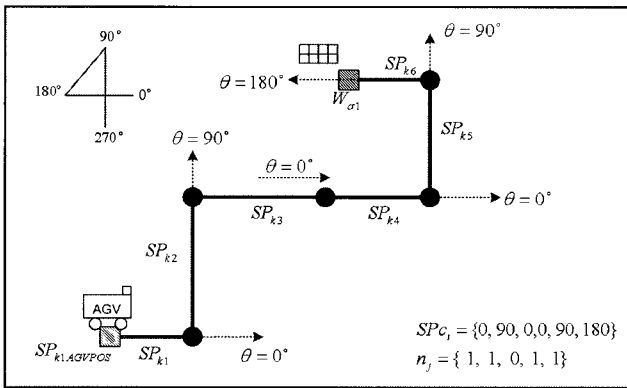


그림 6. SP_k 의 방향(SP_{c_i}) 및 방향 전환 (n_j) 좌표계
Fig. 6. Coordinate system for SP_{c_i} and n_j in SP_k

방향 및 방향 전환 데이터를 이용함으로써 AGV가 주행 중 방향 전환에 따른 감속운전 $\gamma(t)$ 여부를 알 수 있다.

3.4 최단 경로의 경로 별 소요 시간 알고리즘

본 논문에서는 13에서 제안한 AGV의 가감속을 고려한 주행시간 추정 방법을 바탕으로 경로간 방향 전환을 고려한 AGV의 경로 별 소요 시간 산출 방법을 제안한다.

경로별 소요시간은 AGV의 속도 특성에 따라 3가지로 나

눌 수 있다. 첫 번째는 가속이 포함된 최초 경로 주행 시간 (t_1)이고 둘째로 평균속도 주행 및 방향 전환 시 감속운행을 하는 경로 간 주행 시간(t_2), 마지막으로 작업장에서 정지하기 위한 감속이 포함된 최종 경로 주행 시간(t_3)이다. (단, 최단 경로 주행 전 AGV의 최초 속도는 0이라고 가정한다.)

최초 경로 소요시간 t_1 :

$$t_1 = \frac{v}{\alpha} + \frac{P_{14AGVPOS} - dn_1 - \frac{v^2}{2a}}{v} + \frac{dn_1}{r} \quad (7)$$

최초 AGV가 속도 $v(t)$ 까지 이르기 위한 가속 시간($\frac{v}{\alpha}$) + 가속하는데 이동한 거리 ($-\frac{v^2}{2a}$)와 방향 전환 데이터(n_1)에 따른 거리($-dn_1$)를 제외한 $P_{14AGVPOS}$ 거리를 평균 속도 (v)로 주행하는데 걸리는 시간 + 방향전환 데이터 n_1 에 따른 통제구역 거리 d (그림 5 및 표 2 참조)의 감속 운전 시간으로 계산할 수 있다.

경로 간 주행 시간 t_2 :

$$t_2 = \sum_{a=2}^m \sum_{\beta=2}^{m-1} \left(\frac{SP_{a4} - 2dn_{\beta}}{v} + \frac{2dn_{\beta}}{r} \right) \quad (8)$$

각각의 최종 경로 이전까지의 각 경로간의 모든 소요 시간을 구하는 식으로써 $m-1$ 개인 최단 경로의 각 경로 거리에서 각각의 방향 전환 데이터 n 에 따른 통제 구역 거리 d 를 고려한 거리를 속도 v 로 주행한 시간 + 방향 전환데이터에 n 에 따른 거리 d 를 주행한 시간이다.

최종 경로 주행 시간 t_3 :

$$t_3 = \frac{W_{\delta 4} - dn_{m-1} - \frac{v^2}{2\beta}}{v} + \frac{dn_{m-1}}{r} + \frac{v}{\beta} \quad (9)$$

최종 경로 거리 $W_{\delta 4}$ 와 방향 전환 데이터 n_{m-1} 에 따른 거리 d , 정지하기 위해 이동한 거리 $-\frac{v^2}{2\beta}$ 를 고려하여 속도 v 로 이동한 시간 + 마찬가지로 방향 전환 데이터 n_{m-1} 을 고려한 거리 d 를 r 로 이동하는데 걸리는 시간 + 정지하기 위해 드는 시간($\frac{v}{\beta}$)의 합이다. 식 (7)~(9)에서 유도한 경로별 주행 시간 t_1, t_2, t_3 를 바탕으로 시간에 따른 경로별 점유시간을 알 수 있다.

4. 시뮬레이션 및 결과

시뮬레이션에 쓰일 데이터는 Asef-Vaziri[11]가 제시한 작업 환경 데이터를 사용하였다. 작업환경 데이터는 총 총 49개의 경로(P_i)와 34개의 정점(TZ)을 갖으며 그림 6과 같이 도시된다. 작업 환경에서 각 경로별 거리와 ID등은 표 3과 같다. 본 논문에서는 시뮬레이션을 위해 10대의 AGV가 운용중인 상태라고 가정하며, AGV에 할당된 10개의 작업 테이블은 표 4와 같다고 가정한다.

AGV의 가속도 및 속도는 최이 등이 가정한 예를 사용하되 통계구역 거리 d 를 20m로 정의하여 표 5에 나타내었다. 시뮬레이션은 Pentium4-3.2Ghz/1Gbyte RAM, Window XP Pro. SP2의 환경 하에 MATLAB 6.5를 이용하여 실행하였다.

표 3 작업 환경의 각 경로 거리 및 ID
Table 3. Path length and ID of working environment

경로 ID	길이 (m)	경로 ID	길이 (m)	경로 ID	길이 (m)	경로 ID	길이 (m)	경로 ID	길이 (m)
1	450	11	600	21	300	31	250	41	150
2	300	12	150	22	250	32	150	42	200
3	150	13	450	23	450	33	250	43	250
4	600	14	150	24	150	34	150	44	150
5	300	15	150	25	250	35	150	45	150
6	150	16	150	26	150	36	150	46	150
7	300	17	150	27	150	37	300	47	300
8	150	18	150	28	200	38	150	48	450
9	150	19	150	29	250	39	150	49	600
10	150	20	150	30	250	40	300		

표 4. 목표 작업장 ($W_{\sigma 1}$) 및 AGV의 위치 ($P_{\sigma 1, AGVPOS}$)
Table 4. Point of target working area and AGV position

	1	2	3	4	5
$P_{\sigma 1, AGVPOS}$	225,550	750,150	450,625	825,850	1225,400
$W_{\sigma 1}$	1000,400	450,775	300,150	600,275	525,150
경로 위치	11→18	25→6	11→24	3→16	19→25
	6	7	8	9	10
$P_{\sigma 1, AGVPOS}$	1425,400	1425,150	900,775	1225,150	75,150
$W_{\sigma 1}$	750,625	300,150	525,400	1225,850	450,775
경로 위치	20→12	28→24	8→16	28→5	22→6

표 5. AGV의 속도 특성
Table 5. Velocity profile of AGV

$v(t)$	$\alpha(t)$	$\beta(t)$	$\gamma(t)$	d
4m/s	0.64m/s ²	1.55m/s ²	2m/s	20m

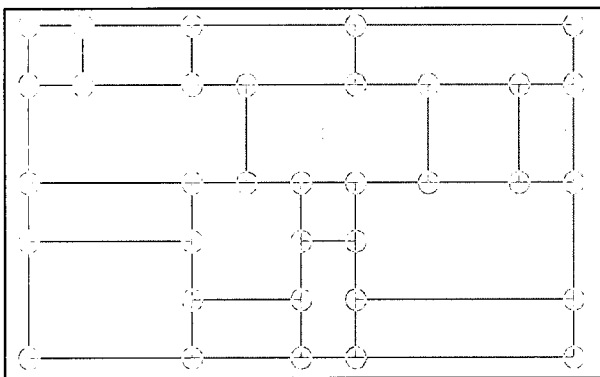


그림 7. AGV의 작업환경
Fig. 7. AGV's working environment

본 논문에서 제안한 평가함수를 이용하여 구현한 최단 경로 탐색알고리즘으로 그림 8과 같이 10개의 작업장에 대한 최단 경로의 결과를 얻을 수 있었다. 또한, 생성된 10개의 최단 경로에 대하여 본 논문에서 제안한 최단 경로 소요 시간 산출 식(7)~(9)를 이용하여 그림 9와 같이 경로 별 소요 시간을 계산할 수 있었으며 최단 경로 방향 및 방향 전환 데이터를 각각 그림 10, 11에 나타내었다.

The screenshot shows a software window titled 'Numeric format: shortG Size: 10'. It displays a 10x10 matrix of numerical values representing the shortest paths between 10 nodes. The values are as follows:

	1	2	3	4	5	6
1	13	14	24	26	27	28
2	40	25	24	14	15	6
3	15	14	24	25	39	37
4	3	8	17	16	26	25
5	30	28	27	26	25	39
6	32	30	28	19	18	17
7	44	43	42	40	39	37
8	9	20	18	16	25	24
9	43	31	32	21	10	4
10	35	22	12	13	15	6

그림 8. 최단 경로 탐색 알고리즘에 의해 생성된 10개의 최단 경로

Fig. 8. 10 shortest paths by using of shortest path algorithm

The screenshot shows a software window titled 'Numeric format: shortG Size: 10'. It displays a 10x10 matrix of numerical values representing the travel times for the shortest paths between 10 nodes. The values are as follows:

	1	2	3	4	5	6
1	64.375	111.88	154.38	191.88	229.38	255.67
2	45.625	118.13	165.63	208.13	245.63	265.67
3	21.875	64.375	111.88	184.38	226.88	265.67
4	26.875	69.375	106.88	149.38	196.88	234.42
5	34.375	84.375	121.88	164.38	236.88	261.92
6	21.875	84.375	139.38	186.88	234.38	259.42
7	21.875	84.375	134.38	209.38	246.88	285.67
8	21.875	64.375	111.88	159.38	201.88	221.92
9	39.375	111.88	159.38	239.38	281.88	356.92
10	26.875	94.375	136.88	259.38	301.88	321.92

그림 9. 10개의 최단 경로에 대한 경로별 소요시간
Fig. 9. Each travel time of 10 shortest paths

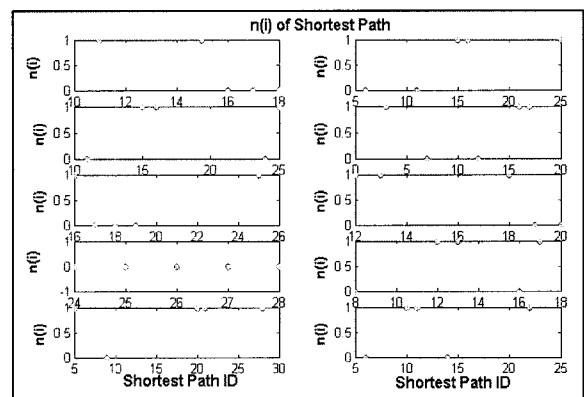


그림 10. 10개의 최단 경로에 대한 방향 전환 데이터(n_i)
Fig. 10. n_i data of 10 shortest path

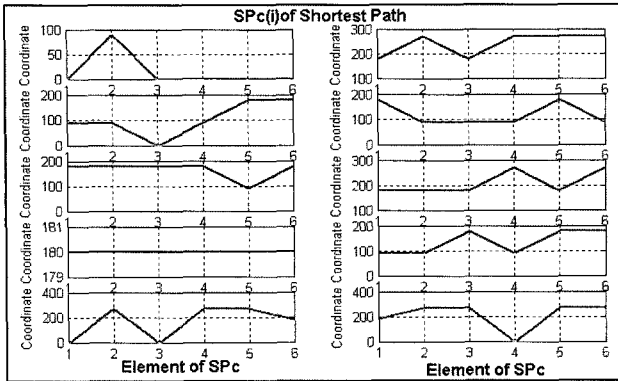


그림 11. 10개의 최단 경로에 대한 방향 데이터 (SPc)
Fig. 11. SPc data of 10 shortest path

5. 결 론

본 논문에서는 장애물을 고려한 경로 선정 기법 및 교차 해결의 방안으로 먼저, 작업 환경에 대한 모델링을 제안하였다. 이를 이용하여 모델링된 작업 요소를 바탕으로 A* 알고리즘의 평가함수 선정 방법 및 작업 환경 요소로서의 그 표현을 제안하였으며 제안된 방법에 의해 최단 경로 탐색 알고리즘을 성공적으로 구현 및 그 결과를 제시하였다. 또한, 생성된 최단 경로를 본 논문에서 제안한 경로 별 소요 시간과 방향 및 방향 전환 데이터로 나타냄으로써 시간에 따른 AGV의 경로 점유시간 및 경로 진입 시간을 예측할 수 있음을 증명하였다. 마지막으로 간단한 실험을 통해 본 논문에서 제안한 방법의 응용 가능성을 증명하였다.

참 고 문 헌

[1] Vis, I.F.A., "Survey of research in the design and control of automated guided vehicle systems". European Journal of Operational Research 170(3), 677-709, 2006, 10.
 [2] Le-Anh T. and Koster M.B., "A review of design and control of automated guided vehicle system". Eur J Oper Res, Vol. 171, pp. 1-23, 2006, 8.
 [3] Qui L, Hsu W.J., Huang S.Y. and Wang H. "Scheduling and routing algorithms of AGVs: a survey". Int J Prod Res, Vol. 40, No. 3, pp. 745~760, 2002, 10.
 [4] Ho T. C.. "A dynamic-zone strategy for vehicle-collision prevention and load balancing in an AGV system with a single-loop guide path". Comput Ind, Vol. 42, pp. 159~176, 2002, 10.
 [5] Hwang H. and Kim S. H.. "Development of operation Rrules for automated guided vehicle systems in heterarchical manufacturing system". Journal of the Korean Institute of Industrial Engineers. Vol. 23, No. 2, 1997, 6.
 [6] Moorthy R.L., Guan W.H., Cheong N.W. and Piaw T.C. "Cyclic deadlock prediction and avoidance for zone-controlled AGV system". Int J

Prod Econ Vol. 83, pp. 24-30, 2003, 6.
 [7] Rajotia S, Shanker K, and Batra J.. "A semi-dynamic time window constrained routing strategy in an AGV system". Int J Prod Res, Vo: 36, No. 1, pp. :35 - 50, 2003, 5..
 [8] George F L., Artificial intelligence structures and strategies for complex problem solving, Addison Wesley Publishers, Fifth edition, 2005.
 [9] Yves C., "Heuristics for large constrained vehicle routing problems". Journal of Heuristics, Vol. 5, pp. 281-303, 1999
 [10] http://en.wikipedia.org/wiki/A*_search_algorithm.
 [11] Ardavan A.Vi, Marc G., "Dual track and segmented single track bidirectional loop guide path layout for AGV systems". European Journal of Operational Research, 2006.
 [12] Ayoub Insa Corréaa, André Langevina and Louis-Martin Rousseaua. "Scheduling and routing of automated guided vehicles: A hybrid approach", Computers & Operations Research, Vol. 34, pp. 1688-1707, 2007, 10.
 [13] 최이, 박대진, 류광렬. "가감속을 고려한 교차없는 AGV 주행경로 설정". 한국해양항만학회지, 제30권 제10호. pp.855~860, 2006, 10.
 [14] Pooya Farahvash, Thomas O. Boucher. "A multi-agent architecture for control of AGV systems" Robotics and Computer-Integrated Manufacturing, Vol. 20, pp. 473-478, 2004, 8.

저 자 소 개

주영훈(Young Hoon Joo)



2007년 : 제 17권 제 4호 참조

김종선(Jong Seon Kim)



2007년 : 군산대 전기전자제어학과 졸업
 2007년~현재 : 동 대학원 전기전자제어공학과 석사과정

관심분야 : Mobile Robot, 신경회로망, 지능제어.
 Phone : 063-469-4706
 Fax : 063-469-4706
 E-mail : roboinside@kunsan.ac.kr