

## 로봇 경로 제어를 위한 속도기반 Dijkstra 알고리즘\*

천성권\*\* · 김근덕\*\*\* · 김종근\*\*\*\*

### A Speed-Based Dijkstra Algorithm for the Line Tracer Control of a Robot\*

Seongkwon Cheon\*\* · GeunDeok Kim\*\*\* · Chonggun Kim\*\*\*\*

#### ■ Abstract ■

A robot education system by emulation based on Web can be efficiently used for understanding concept of robot assembly practice and control mechanism of robot by control programming. It is important to predict the path of the line tracer robot which has to be decided by the robot. Shortest Path Algorithm is a well known algorithm which searches the most efficient path between the start node and the end node. There are two related typical algorithms. Dijkstra Algorithm searches the shortest path tree from a node to the rest of the other nodes. A\* Algorithm searches the shortest paths among all nodes. The delay time caused by turning the direction of navigation for the line tracer robot at the crossroads can give big differences to the travel time of the robot. So we need an efficient path determine algorithm which can solve this problem. Thus, It is necessary to analyze the overhead of changing direction of robot at multi-linked node to determine the next direction for efficient routings. In this paper, we reflect the real delay time of directional changing from the real robot. A speed based Dijkstra algorithm is proposed and compared with the previous ones to analyze the performance.

Keyword : Speed-Based, Dijkstra Algorithm, A-Star Algorithm, Robot, Shortest Path, Line Tracer

논문투고일 : 2011년 07월 22일      논문수정완료일 : 2011년 09월 29일      논문게재확정일 : 2011년 10월 02일

\* 이 연구는 2010년도 영남대학교 학술연구조성비에 의한 것임.

\*\* 가톨릭상지대학 중국비즈니스정보과

\*\*\* 영남대학교 공과대학 컴퓨터공학과

\*\*\*\* 영남대학교 공과대학 컴퓨터공학과 교수, 교신저자

## 1. 서 론

로봇은 어린이부터 어른에 이르기까지 누구나 즐길 수 있는 인터랙티브(Interactive)한 IT기기 기반 창작 및 교육 도구가 될수 있다. 일반적으로 프로그래밍을 처음 배우는 초보자들이 단순한 텍스트처리나 숫자를 계산하면서 고급 프로그래머가 되기까지의 반복학습을 진행하는데, 이는 무척 지루하고 힘겨운 일이다. 반면, 로봇의 동작을 제어하는 프로그램을 하는 것은 그 결과가 로봇의 동작으로 즉시 보이므로 프로그래밍을 배우는 학생들에게 지속적인 동기부여와 성취감을 느끼게 하면서 기술의 수준을 올릴 수 있는 좋은 방법이 된다[3].

일반적으로 경로탐색 로봇의 실습 교육을 하기 위해서는 실습 소프트웨어를 클라이언트-PC에 설치한 뒤, PC와 로봇을 연결하여 명령어를 로봇에 직접 프로그래밍 하는 번거로운 절차가 있다. 또한 자체 제작된 경로지도와 로봇이 이동하기에 충분한 공간과 지정된 장소에서만 실습을 해야 하는 문제점이 있다. 이와 같은 시간적·공간적 제약을 줄이고 경로탐색 로봇의 모든 요소들을 고려한 에뮬레이션 시스템 환경을 구축하고[3], 더 나아가 경로탐색에 있어서 목표점까지 효율적인 최단 경로를 구하는 알고리즘을 연구하여 실습효과를 증진시키는 것이 본 연구의 목적이다.

최단 경로 문제의 도출을 위한 탐색의 실질적인 이론적 탐구는 Dijkstra의 최단경로 탐색기술로 이미 기초가 확립되었고, 이후로 각종 알고리즘에 대한 성능비교와 장애물 회피, AGV 경로, 도로주행 안내 등의 다양한 문제에 응용되고 있다[4]. 그러나 경로탐색에 있어서 단순한 최단경로 보다는 환경상황을 고려한 경로선택이 더 올바를 수 있다. 특히 로봇의 최단 주행시간 비교에 있어서는 직선 주행속도와 회전속도의 차이점에 따른 시간지체를 고려한 지능적 경로 선정이 요구된다.

본 연구에서는 출발노드에서 도착노드에 이르는 경로 중에서 가장 시간 효율적인 경로를 찾는 알고리즘으로 속도기반-Dijkstra 알고리즘을 제안한

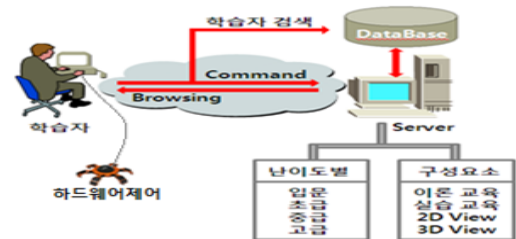
다. 제안된 알고리즘의 실험을 위해 주행 로봇에 물레이션에서 C# 언어를 이용한 경로탐색 인터페이스를 개발하고, 실제 경로탐색 로봇의 여러 가지 형태의 실습용 지도를 기반으로 각 지도의 교차점을 노드로 추출하고 경로 네트워크를 구축하여 테스트 한다.

## 2. 관련 연구와 연구 배경

### 2.1 웹기반 로봇제어 교육 시스템

#### 2.1.1 웹기반 로봇제어 교육 시스템 환경

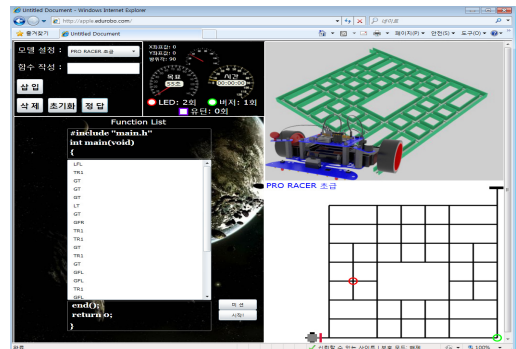
실험환경으로 개발된 주행로봇 시스템에서는 학습자와 교수자가 웹 기반으로 모든 학습 관련 기능에 접근이 가능하다. [그림 1]은 시스템의 동작 환경 및 개념을 보이고 있다[3].



[그림 1] 시스템 동작 환경

#### 2.1.2 주행로봇 제어기능

교육용 및 실험용으로 활용하는 라인트레이서 로



[그림 2] 실습 화면(라인트레이서)

봇은 [그림 2]의 예와 같다. 실제 라인트레이서 로봇은 직진 주행 시의 속도와 회전을 할 경우의 속도가 달라진다. 따라서 최소시간 경로를 탐색하는 경우는 단순한 최단거리가 아닌 속도를 기반으로 하는 최단거리 추출이 필요하다[3].

## 2.2 경로 탐색 알고리즘

경로를 탐색하는 대표 알고리즘으로 A\*(A-Star) 알고리즘과 Dijkstra 알고리즘이 있다. 이 알고리즘들은 계산시간, 사용메모리 등에서 차이가 있다.

### 2.2.1 A\* Algorithm

A\* 알고리즘은 주어진 출발 노드에서 도착 노드까지 가는 최단경로를 찾아내는 그래프 트리 탐색 알고리즘 중 하나이다. 모든 노드들의 검색을 진행하는 상황을 기록하며, 향후 검색 방향을 결정하는데 사용된다[2, 5].

검색된 노드들은 개별적인 노드들의 위치정보뿐 아니라, 목표(Goal), 휴리스틱(Heuristic), 적합도(Fitness)의 3가지 속성을 가지고 있다.

$$f(x) = g(x) + h(x) \quad (1)$$

- 목표(g) : 시작 노드에서 현재 노드 x까지 이동하는데 드는 최단비용
- 휴리스틱(h) : 현재 노드 x에서 목표 노드까지의 예상비용
- 적합도(f) : 목표(g)와 휴리스틱(h)의 합산비용으로써 전체 경로의 비용을 의미하기 때문에 값이 낮을수록 최단경로일 가능성이 높음

A\* 알고리즘은 열린 목록(Open List)과 닫힌 목록(Close List)을 사용한다. 열린 목록은 탐색하지 않은 노드들을 닫힌 목록은 탐색한 노드들로 구성되어 진다.

알고리즘을 정의하면 다음과 같다.

- (1) 시작지점을 열린 목록에 추가

- (2) 열린 목록 노드 중, 1개 추출하여 8방향 주변노드 탐색(평가함수  $F = G+H$  계산과 부모노드 명시)
- (3) 2단계에서 추출한 노드를 닫힌 목록에 삽입
- (4) 2단계에서 탐색한 노드들을 열린 목록에 삽입
- (5) 열린 목록 중 가장 앞 노드를 삭제 후 그 노드를 닫힌 목록에 추가
- (6) 5단계에서 삭제한 노드의 8방향 주요 노드를 탐색
- (7) 열린 목록에 존재하지 않는 노드는 열린 목록에 추가하고, 중복되는 노드는 G값을 서로 비교하여 더 작은 값을 열린 목록으로 교체
- (8) 5단계부터 반복 실행

### 2.2.2 Dijkstra Algorithm

Dijkstra 알고리즘은 어떤 링크도 음수 값을 갖지 않는 방향 그래프에서 주어진 출발 노드와 도착 노드 사이의 최단 경로를 구하는 알고리즘으로서, 매 반복마다 거리의 누적 값이 가장 작은 노드의 경로를 표시하는 Label-setting 기법이다[6-8].

알고리즘을 정의하면 다음과 같다.

- (1) 인접행렬 상태의 각 링크의 가중치(Weight) 표시
- (2) 출발 노드를 초기 값으로 하는 집합 S와 출발 노드를 제외한 모든 노드를 포함하는 집합 T의 초기화
- (3) 출발 노드를 제외한 모든 노드와의 거리 초기 값( $\text{dist}[i], 2 \leq i \leq n$ )을 1단계 인접행렬에서 취함
- (4) 집합 T의 원소 중, 출발 노드부터의 거리가 최소인 노드 v를 택하여 집합 T에서 제거하고 집합 S에 추가
- (5) 집합 T의 모든 정점 w에 대해 출발 노드부터의 거리( $\text{dist}[w]$ )와 간선 E(v,w)의 길이에 v정점의 거리( $\text{dist}[v]$ )를 합한 값 중, 작은 것을 선택하여 정점 w의 거리 선택

(6) 4단계와 5단계의 과정을 집합 T가 공집합이 될 때까지 반복 실행

2.2.3 A\*와 Dijkstra 알고리즘 비교 · 분석

A\* 알고리즘은 최단경로를 구축하는 과정에서 현재 노드에서 도착 노드까지의 잔여경로비용의 최소추정치를 이용하는 방법으로 추정된 값이 최소인 방향으로만 탐색이 일어나도록 하는 것이다. 주로 그래프의 경로비용이 거리인 형태의 잔여 경로비용을 추정하기 쉬운 그래프에 적용하는 알고리즘이다.

Dijkstra 알고리즘은 그래프 이론의 하나로 어떤 링크도 음수 값을 갖지 않는 방향 그래프에서 주어진 출발 노드와 도착 노드 사이의 최단 경로 문제를 푸는 알고리즘으로 응용 및 분석이 용이하다.

본 연구에서는 회전속도에 따른 시간지체의 고려유무에 따라 탐색된 경로의 성능을 비교 · 분석하기 위해 기준이 되는 알고리즘으로 Dijkstra 알고리즘을 기준으로 한다.

3. 속도기반-Dijkstra 알고리즘

경로탐색 로봇의 이동네트워크에서는 이동구간의 주행시간에 비하여 교차로에서의 방향별 회전으로 인하여 발생하는 통행시간이 로봇주행시간에 큰 영향을 끼친다고 판단하여 이를 고려한 경로탐색이 필요하다. 따라서 실제 로봇의 회전에 따른 지체시간을 고려한다.

기본이 되는 Dijkstra 최단 경로 탐색 알고리즘에서 각 링크의 거리를 시간으로 환산하고 Dijkstra 알고리즘과 다르게 각 노드의 값은 방향전환에 따라 처리시간이 달라진다.

총 주행시간(sec) = 실 거리 주행시간+회전 지연시간 (2)

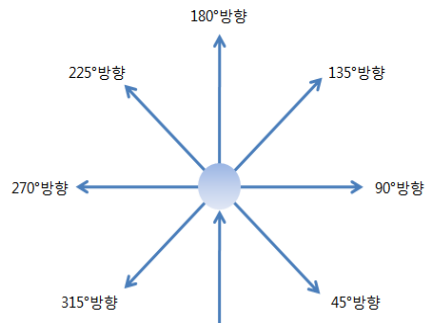
• 실 거리 주행시간 =  $\frac{\text{전체 주행거리}}{\text{경로탐색 로봇 속도}}$  (3)

• 회전 지연시간 =

$\sum NR_{ij} \times \text{경로탐색로봇속도}$  (4)

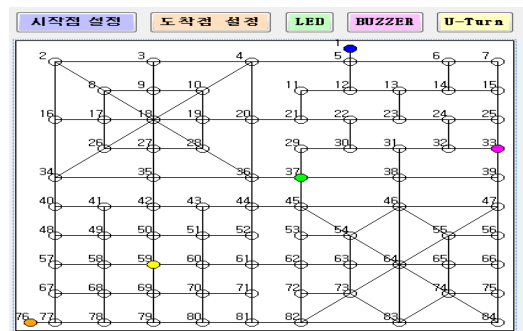
• NR(Node Rotation) = 각 정점 회전유무에 따른 지연시간

본 연구에서는 [그림 4]와 같은 라인트레이사용 경로의 각 노드에서 만날 수 있는 방향전환에 대해 고려한다. 하나의 노드에서 결정할 수 있는 방향은 [그림 3]과 같이 8가지로 각 방향에 대해 회전지체시간이 달라진다. [그림 3]에서 되돌아가는 경우는 없으며 90° 방향, 135° 방향, 180° 방향은 다른 지연 값을 가진다. 90° 방향과 270° 방향은 같은 지연 값을 가진다.



[그림 3] 8가지 다른 속도 값을 가지는 방향전환

3.1 시작 · 도착 노드 설정 및 미션 설정



[그림 4] 시작 · 도착 노드 설정 및 미션 설정

[그림 4]는 알고리즘을 실행하기 위한 선행단계로써, 시작노드와 도착노드를 설정하고 경로탐색

로봇이 수행해야 할 미션 3가지(LED, BUZZER, U-Turn)를 설정해주는 단계이다.

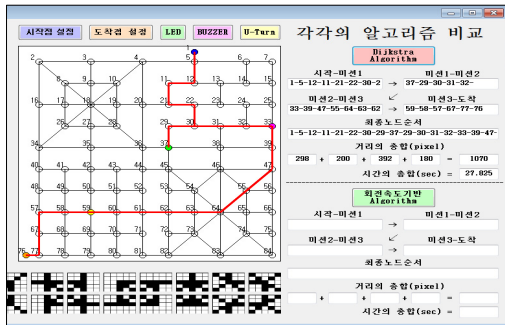
시작·도착 노드 설정 및 미션 설정의 하나의 예로서, 다음과 같이 임의의 위치에 적용하여 설명할 수 있다.

- (1) 시작노드 : 1번, 도착 노드 : 76번,
- (2) 미션(LED, BUZZER, U-Turn) : 37번, 33번, 59번

### 3.2 Dijkstra알고리즘 실행

[그림 5]는 시작 노드와 도착 노드 사이의 각 미션을 수행 할 때, Dijkstra 알고리즘을 기반으로 최단경로가 설정된 그림이다.

- (1) 최종 최단경로 : 1 → 5 → 12 → 11 → 21 → 22 → 30 → 29 → 37 → 29 → 30 → 31 → 32 → 33 → 39 → 47 → 55 → 64 → 63 → 62 → 61 → 60 → 59 → 58 → 57 → 67 → 77 → 76(Red Line)
- (2) 거리의 총합(pixel) : 1070
- (3) 시간의 총합(sec) : 27.825



[그림 5] Dijkstra 알고리즘

## 4. 경로탐색 시스템 환경 구축과 실험

경로탐색을 위한 에뮬레이션 시스템은 대상이 되는 경로 패턴을 노드와 링크로 구성되는 네트워크에서 추출한다. 이를 위해서 크게 3가지 주요 기

능으로 구성되어 있다.

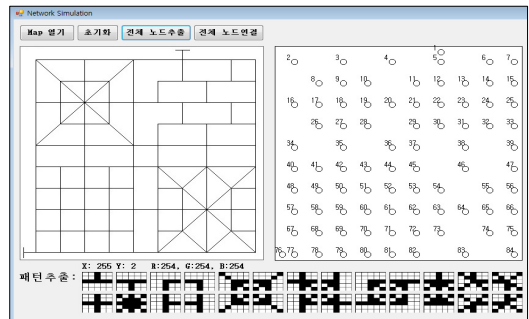
- (1) 각 교차로의 노드화 및 패턴 추출
- (2) 각 노드들의 패턴에 맞는 링크 연결
- (3) 시작·도착 노드 설정 및 미션 설정

대상 네트워크에서는 다음의 2종류의 알고리즘을 실행한다.

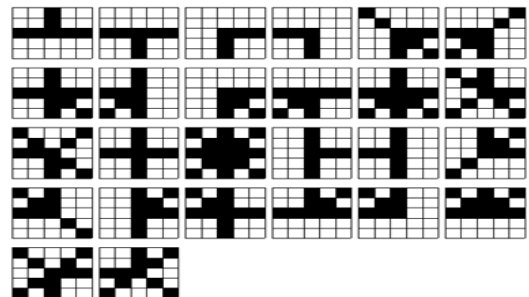
- (1) Dijkstra 알고리즘 실행
- (2) 속도기반-Dijkstra 알고리즘 실행

### 4.1 각 교차로의 노드화 및 패턴 추출

[그림 6]은 인식된 400×400(pixel)크기의 경로탐색 지도를 분석하여 각 교차로를 노드화한 뒤, 노드 번호를 할당한 형태이다.



[그림 6] 노드화 및 패턴 추출



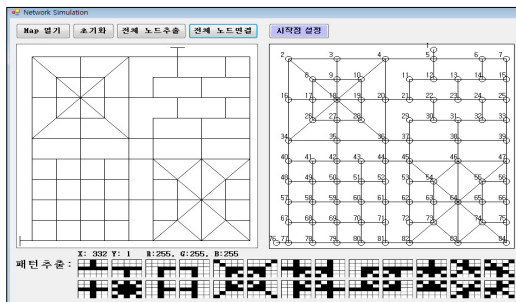
[그림 7] 5×5(pixel)크기의 링크와 연결된 노드 패턴추출

각 교차로를 노드화하기 위해서는 우선적으로 노드의 패턴을 추출해야 한다. 5×5(pixel)크기의 마스

크를 기준으로 패턴에 맞는 정점을 추출하여 저장하고, 추출된 순서대로 노드번호를 삽입한다. [그림 7]은 5×5(pixel)크기의 마스크를 이용하여 노드에 연결된 링크정보까지 포함하는 패턴을 추출한 결과이다.

### 4.2 각 노드들의 패턴에 맞는 링크 연결

[그림 8]은 추출된 노드 패턴에 맞게 노드와 노드사이의 링크를 연결한 형태이다. 노드와 노드사이를 연결하기 위해서는 각 노드에 연결된 링크 방향성을 판단할 수 있어야 한다. 각각의 노드 점에서 8방향 pixel별로 선의 연결유무를 확인한 다음, 방향이 결정되면 그 방향으로 노드 간 최대한 격만큼 설정하여 검색하면서 특정 위치(교차로 위치)와 매칭이 되면 노드 간 링크를 연결한다.



[그림 8] 패턴에 맞는 링크 연결

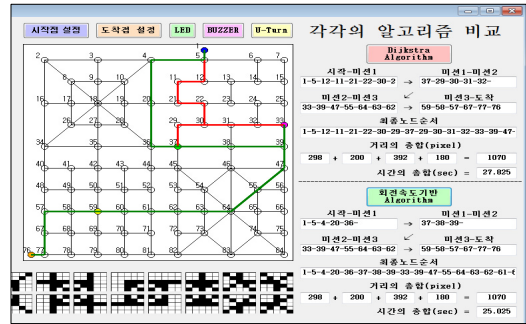
### 4.3 속도 기반-Dijkstra 알고리즘 실행

속도 기반-Dijkstra 알고리즘은 주행 로봇이 방향을 바꿀 때 발생하는 속도 지연을 고려하여 최단 경로를 결정한다.

[그림 9]는 시작노드와 도착노드 사이의 각 미션을 수행함에 있어서 회전속도를 고려한 Dijkstra 알고리즘을 토대로 최단경로가 설정된 그림이다.

- (1) 최종 최단경로 : 1 → 5 → 4 → 20 → 36 → 37 → 38 → 39 → 33 → 39 → 47 → 55 → 64 → 63 → 62 → 61 → 60 → 59 → 58 → 57 → 67 → 77 → 76 (Bold Line)

- (2) 거리의 총합(pixel) : 1070
- (3) 시간의 총합(sec) : 25.025

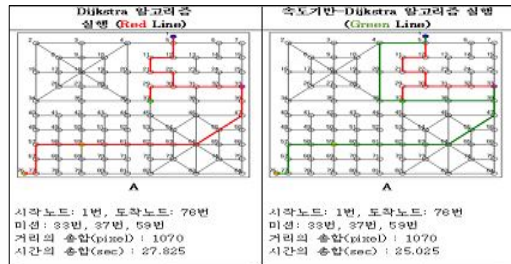


[그림 9] 속도 기반-Dijkstra 알고리즘 적용 경로

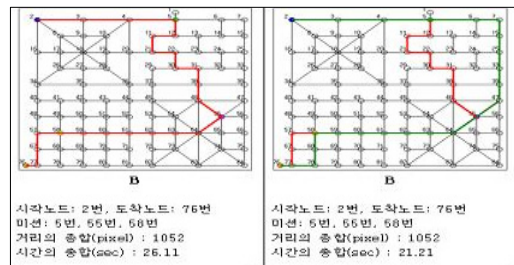
### 4.4 비교 · 분석 및 성능평가

#### 4.4.1 알고리즘 비교 · 분석

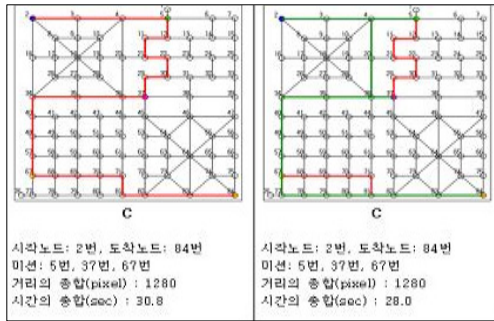
다양한 미션을 부여하여 Dijkstra 알고리즘과 속도기반-Dijkstra 알고리즘을 적용한 결과를 [그림 10]에 보인다. 그림 A, B, C는 서로 다른 경로 패턴에서 알고리즘 수행 결과를 보이며, 모두 정상적으로 수행이 되었음을 알 수 있다.



[그림 10](A)



[그림 10](B)



[그림 10](C)

[그림 10] 다양한 미션에서의 각 알고리즘 적용 예

#### 4.4.2 알고리즘 성능평가

제한한 경로설정 알고리즘의 효율성을 검증하기 위해서 A\* 알고리즘, Dijkstra 알고리즘, 속도기반-Dijkstra 알고리즘의 세 가지 방법으로 경로탐색을 시행하여 이를 비교·분석한다.

비교를 위해 세 가지 탐색 알고리즘에 적용한 이동시간은 산정된 거리에 대한 속도의 지체를 고려한 이동시간으로 환산한 값을 이용한다. 해당 분석에서 활용한 이동거리는 실제 경로탐색로봇의 지도를 비례축소 제작하였으며, 통행시간 또한 실제 이동시간을 통해 도출한 링크 통행시간을 사용하였고, 회전 시 적용한 시간의 지체는 앞서 제시한 식 (1)~식 (3)을 통하여 도출된 값을 사용하였다.

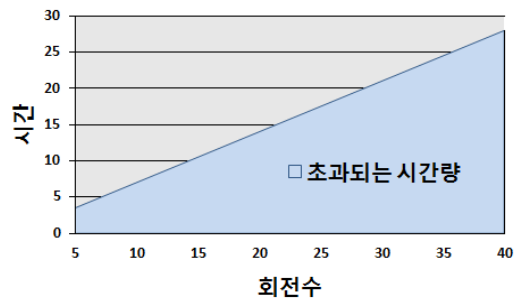
<표 1>은 시작 노드에서 도착 노드까지의 서로 다른 9개 지도의 경로를 3가지 알고리즘을 이용하여 탐색한 결과의 경로 거리와 소요 시간을 보인다.

거리와 시간은 짧은 것이 효율적임을 나타낸다.

거리 면에서 A\* 알고리즘 평균 1030.7픽셀, Dijkstra 알고리즘 평균 1016.4픽셀, 속도기반-Dijkstra 알고리즘 1016.4픽셀로 도출되었으며, A\* 알고리즘이 다른 알고리즘에 비해 평균적으로 14.3픽셀 더 길게 나타났다.

시간 면에서는 A\* 알고리즘 평균 27.59초, Dijkstra 알고리즘 평균 25.33초, 속도기반-Dijkstra 알고리즘 평균 22.22초로 도출되었으며, 속도기반-Dijkstra 알고리즘이 A\* 알고리즘에 비하여 거리 면에서 약 1.4%, 시간 면에서 약 20%의 감소효과가 나타났다. 또한 속도 기반-Dijkstra 알고리즘은 기준이 된 Dijkstra 알고리즘에 비하여 약 12%의 시간이 감소되는 효과가 도출된 것으로 분석되었다.

[그림 11]은 로봇 회전에 따른 추가 시간을 고려하여 회전수가 많아질수록 정비례하게 통행시간이 증가되는 비율을 보인다.

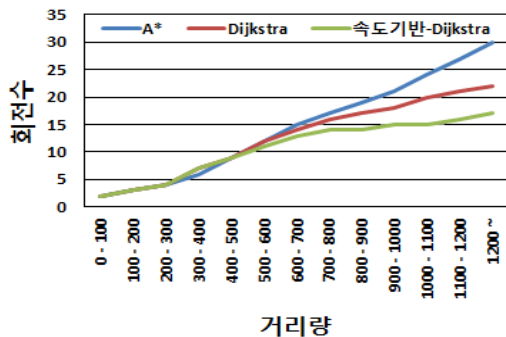


[그림 11] 회전수에 따른 초과되는 시간 량(sec)

<표 1> 경로탐색 실험 결과 평균

알고리즘	지도	A	B	C	D	E	F	G	H	I	평균
	A* 알고리즘	거리(pixel)	1118	1052	1280	1040	1176	852	964	900	894
	시간(sec)	35.665	26.11	35.0	27.3	31.08	24.01	23.17	24.15	21.805	27.59
Dijkstra 알고리즘	지도	A	B	C	D	E	F	G	H	I	평균
	거리(pixel)	1070	1052	1280	1040	1176	852	932	900	846	1016.4
	시간(sec)	27.825	26.11	30.8	24.5	31.08	24.01	21.91	22.75	19.005	25.33
속도기반-Dijkstra 알고리즘	지도	A	B	C	D	E	F	G	H	I	평균
	거리(pixel)	1070	1052	1280	1040	1176	852	932	900	846	1016.4
	시간(sec)	25.025	21.21	28.0	21.7	26.18	19.11	20.51	19.25	19.005	22.22

[그림 12]은 세 가지 탐색 알고리즘에서 거리에 비례한 회전횟수를 도식화 한 것으로, 거리가 멀어질수록 경로에서 회전수가 많아진다. 이 경우 회전속도를 고려한 Dijkstra 알고리즘이 타 알고리즘에 비하여 회전수가 적게 나타나는 것을 알 수 있다.



[그림 12] 탐색방법별 거리비례 평균회전수

특히 경로상에 교차로가 많은 지도일수록 속도 기반-Dijkstra 알고리즘을 이용한 경로탐색기법이 타 기법에 비하여 경로통행시간을 단축시키는 결과를 보인다.

## 5. 결론 및 향후 연구과제

본 논문에서는 경로탐색 로봇에서 기본적으로 필요한 제어기술인 동작제어, 센서, 모터제어와 같은 로봇의 구동중심 알고리즘에 더하여 주행 시간을 줄일 수 있는 효율적인 경로 탐색방안을 연구하였다. 기존 탐색 알고리즘에 더하여 주행 로봇에 적용 가능한 효율적인 최단 경로를 탐색하는 속도 기반-Dijkstra 알고리즘을 제안하였다.

다양한 경로를 가지는 실험 대상 네트워크에서 A\* 알고리즘, Dijkstra 알고리즘, 속도 기반-Dijkstra 알고리즘의 세 가지 방법으로 경로탐색을 수행하여 성능을 비교·분석하였다. 분석 결과 속도 기반-Dijkstra 알고리즘을 이용한 경우 평균 경로통행 시간이 가장 적게 나타났다.

특히 A\* 알고리즘과 비교하였을 경우, 경로의 거리가 커질수록 회전수의 차이가 크게 나타났으며 이는 곧 소요되는 시간이 커짐으로 인해 불합리한 경로를 탐색할 수도 있다는 것을 확인하였다. 따라서 제안된 알고리즘을 경로탐색 로봇의 최단 경로 탐색에 적용하면 시스템의 성능을 향상시킬 수 있다.

해당 연구의 결과는 주행 로봇 및 자동차 경로탐색에서 하나의 방안으로 활용이 가능할 것으로 판단된다. 추후 본 연구에서 사용한 속도기반-Dijkstra 알고리즘을 적용할 수 있는 실용적인 주행 시스템의 연구도 필요할 것이다.

## 참고 문헌

- [1] 심충섭, 김진석, “One-to-One 최단경로 알고리즘의 성능 평가”, 『정보과학회논문지』, 제29권, 제11호(2002), pp.634-639.
- [2] 민근홍 외, “신호 교차로에서 방향별 지체를 고려한 최적경로탐색 연구”, 『한국ITS학회 논문지』, 제9권, 제3호(2010), pp.12-19.
- [3] 김근덕 외, “웹 기반 로봇제어 교육 시스템”, 『한국정보과학회 학술심포지엄논문집』, 제4권, 제1호(2010), pp.35-39.
- [4] 김용민, 최인찬, “로봇을 위한 위치 인식 및 경로 안내 시스템에 관한 연구”, 『전자공학회논문지』, 제29권, 제1호(2010), pp.12-21.
- [5] Zhipeng Liu, “A Study of Routing Protocols for Ad Hoc Networks”, Yeungnam University, Masters of Engineering, 2008.
- [6] Hart, P. E., N. J. Nilsson, and B. Raphael, “A Formal Basis for the Heuristic Determination of Minimum Cost Paths in Graphs”, *IEEE Trans. on Systems Science and Cybernetics*, Vol.SSC-4, No.2(1968), pp.100-107.
- [7] Dijkstra, E. W., “A Note on Two Problems in connexion with Graphs”, *Numerische*



- 
- Mathematik*, Vol.1(1959), pp.269-271.
- [8] Cherkassky, B. V., A. V. Goldberg, and T. Redzik, "Shortest Paths Algorithms : Theory and Experimental Evaluation", *Mathematical Programming*, Vol.73, No.2(1996), pp. 129-174.

## ◆ 저 자 소 개 ◆

**천 성 권 (skcheon@daum.net)**

영남대학교 공과대학 전자공학과 졸업 후 대우중공업에서 근무, 영남대학교 대학원 전산공학과 석사과정 후 동 대학 컴퓨터공학과 박사 수료 후 현재는 가톨릭상지대학 중국비즈니스정보과 재직 중이며, 관심분야는 인터넷 기반 모바일정보시스템, 유무선 네트워크, 센서 네트워크 시스템 등이다.

**김 근 덕 (apple83@ynu.ac.kr)**

영남대학교 이과대학 통계학과 졸업 후 동 대학 컴퓨터공학과 석사수료 후 현재는 삼성전자에서 재직 중이며, 관심분야는 컴퓨터 네트워크, 모바일기반 정보 시스템, 원격 로봇 제어 시스템 등이다.

**김 종 근 (cgkim@yu.ac.kr)**

영남대학교 공과대학 전자공학과 졸업 후 동대학 대학원 전자공학과 계산기전공 석사 수료하였고, (일본)전기통신대학 정보공학과 박사 수료 후, (미국) Virginia Tech. 연구교수, 영남대학교 전산정보원장, (미국) UCSC 연구교수 등을 거친후 현재는 영남대학교 컴퓨터공학과 교수로 재직 중이며, 관심분야는 분산시스템, 컴퓨터통신 및 인터넷, 차세대 인터넷 기술, Ad Hoc 네트워크, 가상강의시스템, 모바일 원격 제어 시스템 등이다.