

Kalman Filter와 Space Syntax를 이용한 GIS 기반 다중경로제공 시스템 개발

Development of the Multi-Path Finding Model Using Kalman Filter and Space Syntax based on GIS

류 승 규

(서울시립대 교통공학과, 석사과정, uos1004@uos.ac.kr)

이 승 제

(서울시립대 교통공학과, 부교수, sjlee@uos.ac.kr)

목 차

I. 연구 개요

1. 연구의 목적 및 범위
2. 기존 연구 및 문헌 고찰

II 다중경로제공 시스템 개발

1. Axialmap 구축
2. 경로탐색 알고리즘

III. 프로그램 구현

1. 경로탐색 Code 흐름
2. A* boundary 내의 경로 탐색 알고리즘의 개선
3. 프로그램 구현

IV. 결론 및 향후 연구과제

참고문헌

I. 연구 개요

기존의 최적경로 알고리즘은 거리, 시간, 교통 통행량 등의 통행값을 고려하여 구성되었다. 하지만 이렇게 제시된 최적경로는 사용자의 인지도에 대한 고려를 하지 않음으로써 통행에 대한 완벽한 정보를 갖지 않는 사용자가 처음으로 통행하는 도로를 접할 때 또는 다수의 사용자가 선호하는 경로를 선택하지 않는 문제점을 가진다. 따라서 본 연구는 Space Syntax 이론을 적용하여 거리, 시간 등을 고려하면서 사용자가 선호하는 경로를 제공할 수 있도록 최적경로 알고리즘을 개발하였다.

1. 연구의 목적 및 범위

본 Space Syntax 이론을 적용한 최적경로 알고리즘 개발을 위해서 본 연구는 첫째로, 대규모 네트워크에 적용 가능한 알고리즘을 개발하는 것이다. 둘째로 Kalman Filtering에 기반을 둔 통행시간 예측 시스템을 GIS 시스템에 적용하여 사용자에게 시간 정보 경로를 제공하는 것이다. 마지막으로 인간이 공간을 인지하고 사용하는 공간 사용행태에 대한 이해에 분석의 기본을 두고 있는 Space Syntax 이론을 적용하여 운전자가 쉽게 인지할 수 있는 경로제공 시스템을 개발하여 찾아가기 쉬운 경로를 제공하는 것이다.

즉, 본 연구는 단순한 비용함수를 사용한 것이 아닌 Space Syntax 이론을 적용한 최적경로 탐색 알고리즘을 개발함으로써 사용자에게 인지도 높은 경로를 제공하여 현실성을 높이고, Stochastic Process를 적용한 알고리즘을 개발함으로써 정확한 통행시간을 제공할 수 있도록 한다.

2. 기존 연구 및 문헌 고찰

기존의 개발된 최적경로 탐색 알고리즘에 대해서는 Traffic Assignment Techniques, Thomas(1991)의 문헌에 잘 정리되어 있으며 교통분야에 가장 보편적으로 많이 응용되고 있는 Dijkstra의 알고리즘, Moore의 알고리즘, 그리고 D'Esopo 알고리즘이 있다. 또한 네트워크가 커질수록 최적경로 탐색 시간이 증가하는 단점을 보완하기위해서 자료구조 알고리즘을 사용하며 L-deque 또는 Bucket을 사용하는 것이 가장 빠른 것으로 조사(F.Benjamin Zhan,1997)되었다. 또한 경로탐색시 일정한 영역을 제한하며 수행속도를 향상시키는 방안이 있으며 대표적인 알고리즘으로 A*알고리즘을 사용한다.

구간 통행시간 예측을 위하여 지금까지 사용한 대표적인 방법은 통계적 기법을 이용하는 것이다. 이들 기법중에는 다중회귀분석, 시계열분석, Kalman Filter, 신경망 이론이 있으며 이승제(2004)의 연구에 의하면 단기통행시간 예측으로서는 Kalman Filter가 가장 적합한 알고리즘으로 분석되었다.

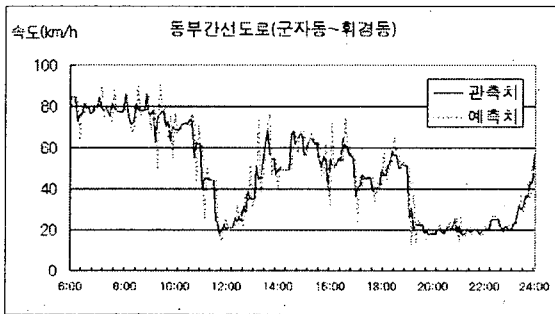
Space Syntax 방법론은 공간구조를 분석하여 각 공간의 속성을 정량적으로 제시하는 이론이자 이를 토대로 개발된 일련의 컴퓨터 프로그램을 칭한다. 본 방법론은 1980,1990년대에 걸쳐 영국 런던대학교의 Hillier 연구팀이 개발하였다.

위 이론은 공간구조상 중요도를 분석대상지역의 전체 공간에서의 접근성에 의하여 계산한다. 즉, 분석대상 범위내의 모든 공간이 기점이자 종점이 되는 가정아래 각 공간의 접근성을 분석하게 된다. 이 접근성을 Space Syntax에서는 전체 공간을 통합하여 주는 통합성 혹은 공간구조상의 위계성을 의미하는 통합성(Integration: 인티그레이션)이라 정의한다. 따라서 Space Syntax 분석 결과에 의한 통합성이 큰 공간은 다른 모든 공간으로부터의 접근성이 양호하다는 것을 의미한다.

II 다중경로 제공 시스템 개발

1. 통행시간 예측

Kalman Filtering은 무작위 변동(random perturbation)을 발생시키는 동적 체계(dynamic system)에 적용되는 최적상태 예측과정으로써 이산적 실시간에 얻어진 잡음 섞인 정보로부터 동적 체계의 미지 상태에 대한 최적 예측을 위한 선형, 불편(unbiased), 최소오차분산(minimum error variance)의 특성을 지닌 반복적 알고리즘이다. 기존 연구에 의하면 단기 통행시간 예측모형으로 많은 예측방법 중에서 Kalman Filtering에 의한 방법이 가장 뛰어난 것으로 나타났다. 따라서 본 연구에서는 최적경로중의 하나로 Kalman Filtering에 의해서 예측된 통행시간을 통하여 경로를 제공하였다.



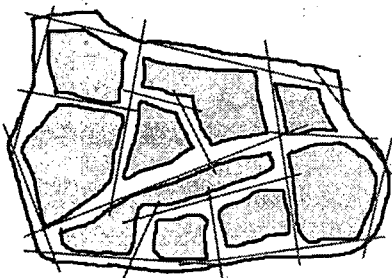
<그림 1> Kalman Filter에 의한 통행시간 예측

2. Axialmap 구축

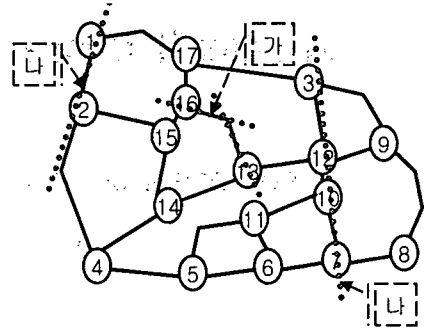
기존의 Space Syntax 분석 프로그램은 Axial Map을 입력하면 분석이 가능하지만 사용자가 Network에 근거한 Axial Map을 일일이 수작업으로 작성해야 하는 단점이 있었다. 또한 대규모 Network에 대해 Axial Map을 작성하는 것은 많은 시간이 소요되고, 분석하는 개인의 주관이 개입되기 쉬운 단점도 있었다. 따라서 Network를 입력으로 Axial Map을 컴퓨터가 자동으로 구성하도록 하는 방법을 고안하고 개발하여 본 연구에 적용하였다.

1) Space Syntax의 Axial Map과 GIS 데이터의 연계 구축 엔진 개발

Space Syntax의 Axial Line <그림 2>와 교통 Network는 <그림 3>는 서로 대응되지 않는다. 실제의 도로망과 Axial Map은 양쪽 모두 수학적인 Graph로 모델을 추상화를 할 수 있으나 이 두 개의 모델에서 node와 link가 의미하는 바는 서로 다르다.



<그림 2> 수작업으로 그려진 Axial Map



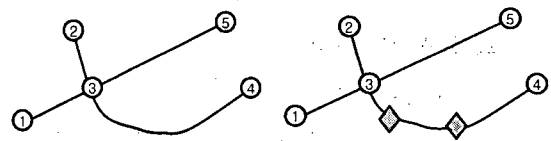
<그림 3> GIS 교통망의 Graph 표현

<표 2> Axial Line과 도로의 대응관계

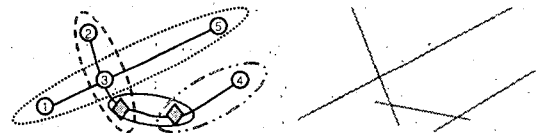
유형	설명	예제
가	단일 도로가 다수의 Axial Line으로 구성되는 경우	16번과 13번 노드를 잇는 링크는 2개의 Axial Line임
나	단일 Axial Line이 다수의 도로에 대응되는 경우	3번과 12번 노드를 잇는 링크, 12번과 10번을 잇는 링크, 10번과 7번을 잇는 링크 이렇게 3개의 링크가 하나의 Axial Line이 됨
다	인접한 링크의 일부 분이 서로 모여 Axial Line을 이룸	1번과 2번을 잇는 링크는 2번과 4번을 잇는 링크의 일부분과 함께 하나의 Axial Line을 구성함

2) Axial Map을 자동으로 구성하기 위한 Algorithm 개발

GIS 데이터로 표현된 도로 Network와 Space Syntax의 Axial Map은 서로 다르지만, GIS 데이터의 링크를 직선으로 나눈 후, 연결된 링크 중 기울기가 비슷한 링크들을 모아 고유번호 (Space Syntax Index)를 부여하는 방법으로 Space Syntax의 Axial Map을 구현할 수 있다. 따라서 다음과 같은 방법으로 도로 Network에서 Axial Map을 구성하는 방법을 개발하였다.



(a) Graph로 표현된 도로망 (b) 곡선링크에 더미노드 추가



(c) 기울기가 비슷한 그룹별로 분류

(d)자동으로 구성된 Axial Map

<그림 4> Axial Map을 자동 구성하는 방법

- 도로 Network에서 각 링크가 직선인지 곡선인지 판단
- 곡선링크일 경우 적절한 '더미 노드'를 추가하여 다수의 링크로 분할

- 추가된 Node를 포함하여, 기울기가 비슷하고 연결된 링크 집합을 구함
- 구해진 각각의 집합에 고유의 Axial Line ID를 부여
- 각각의 Axial Line ID(Space Syntax Index)에 근거하여 상호 접근성을 측정하여 Axial Map을 산출

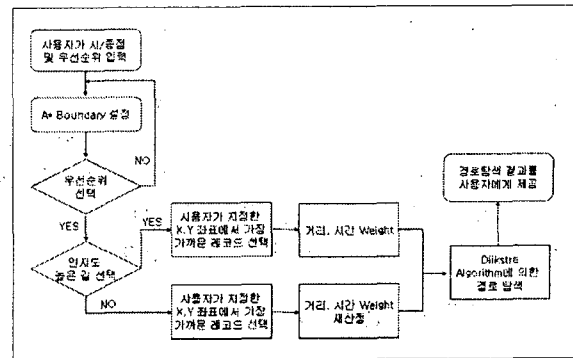
GIS에서 곡선 링크는 사실상 연속된 점들의 집합으로 이루어지고, 이웃한 두 점은 직선이다. 따라서 곡선 링크는 몇 개의 분할된 작은 직선들로 이루어지지만, 이 곡선 링크를 어디서부터 어디까지 하나의 Axial Line으로 볼 것인가 하는 문제가 제기된다. Axial Line을 최대한 긴 직선으로 최소한의 개수가 되도록 그리라는 지침이 있지만 작업일 경우, 일관성 없는 결과를 낳게 된다. 곡선 링크를 몇 개의 Axial Line으로 볼 것인가 하는 것은 주관적인 판단에 의해 변할 수 있으므로, 프로그램에 적당한 파라미터를 입력함으로써 동일한 기준으로 Axial Line의 개수를 객관적으로 판단할 수 있는 방법이 중요하다. 본 연구에서는 이 문제를 해결할 수 있는 몇 가지 방법을 제시하였다.

- 기울기의 변화 이용
 - 인접한 세 점이 직선에 근접할 경우 두 점을 잇는 직선의 기울기는 비슷함
- 인접한 좌표의 삼각형의 넓이 이용
 - 인접한 세 점이 직선에 근접할 경우 세 점을 이루는 삼각형의 넓이는 좁음
- 도로 폭 및 시선 축 이용
 - 도로의 좌표 외에 GIS 도로 속성 가운데 존재하는 도로 폭에 관한 정보를 가지고 Axial Line을 판단하는 방법을 사용가능
 - 도로의 좌표는 중앙선을 따라 격인 부분에만 부여되어 있음. 따라서 도로 폭이란 변수가 있으면 실제의 도로 모습을 재현할 수 있음
 - 도로의 폭을 고려하였을 경우, 가장 긴 직선이 시각적으로 도로 위를 벗어나는지에 따라 세 점으로 Axial Line을 구성할지를 결정

본 연구에서는 Axial Map 생성을 자동화하기 위해 Dijkstra Algorithm을 변형하여 링크를 분류한다. 일반적으로 Dijkstra algorithm은 최단거리 산출에 사용되는 알고리즘이나 인접링크와의 기울기 차이를 cost로 할 경우, 경로탐색과 유사한 방법으로 연결된 최대길이의 Axial Line을 산출할 수 있다.

3. 경로 탐색 알고리즘

사용자가 출발점과 도착점을 지정하고 가장 짧은 거리를 가지는 경로, 시간이 가장 적게 걸리는 경로, 거리를 AxialLine의 통합도로 나누어서 통합도가 높은 경로를 사용자가 선택하고 출발점과 도착점을 입력하면 경로탐색시스템은 사용자가 지정한 X, Y좌표에서 가장 가까운 레코드를 선택하고, 결정점(Node)을 검색해서 최단경로를 찾아준다.



<그림 5> 경로 탐색 흐름도

1) Dijkstra 알고리즘

Dijkstra 알고리즘은 Label-Setting 기법을 적용한 알고리즘으로, 우선 초기 출발노드를 제외한 모든 노드의 표지를 ∞ 로 설정한 다음 초기 출발노드에서 연결된 모든 노드의 표지에 각각의 거리를 입력한다. 이때 입력된 누적거리 값이 가장 작은 노드를 그 다음 노드로 설정하고, 그 노드에 영구표지를 표시한다. 다시 다음 노드에서 연결된 모든 노드를 찾아내어, 각각의 노드 표지에 출발노드로부터의 거리의 누적 값을 입력한다. 다시 그때까지 입력된 표지 값 중 가장 작은 노드를 다음 노드로 하여 위의 과정을 모든 노드에 영구표지가 될 때까지 반복한다.

기존의 최적경로탐색 정보제공시스템은 현재의 교통상황만을 반영하여 교통정보를 제공해 줌으로써 실제의 통행 시 운전자가 경험하는 실제 통행시간과 통행경로에 차이가 발생한다. 교통망은 교통량이 증가하면서 링크의 여행시간은 증가한다. 어느 링크에 교통량이 증가하면 다른 링크가 여행시간이 줄어서 이 링크가 최적경로로 변화할 수 있다. 이러한 사항을 고려하여 최적 경로를 탐색하는 것이 동적 최적 경로 탐색이라고 할 수 있다.

경로로 제공되는 최단거리 경로, 최단 시간 경로(최적 경로), 인지하기 쉬운 경로 등에 따라 알고리즘에 사용되는 통행 비용이 다르게 적용된다. 최단 거리 경로의 경우는 거리를 통행으로 사용하여 Dijkstra 알고리즘에 의한 거리가 최소화 되는 경로를 제공하며, 최단 시간 경로의 경우 Kalman Filter에 의해서 예측된 통행시간을 비용으로 사용하여 통행시간이 최소화 되는 경로를 제공한다. 인지하기 쉬운 경로는 Space Syntax에 의해서 기종점간 접근성이 가장 높은 도로를 포함하여 통행 비용이 최소가 되는 경로를 제공하게 된다.

또한, 현실적으로 Syntax가 반영된 경로의 제공을 위하여 안내거리, 시간 뿐만아니라 Axial Map을 자동구성해서 인지도 높은 Link의 길이를 해당 Link에 속하는 AxialLine의 통합도로 나누어서 통합도가 높은 경로를 본 연구에 적용하였다.

<표 3> 본 연구에 제공된 다중경로

	k(i)
거리	node i까지의 거리
시간	node i까지의 통행시간
거리/통합도	node i까지의 거리+통합도
시간/통합도	node i까지의 시간+통합도

거리는 노선(Link)가 가지는 길이의 총합을 말한다. 시간은 Kalman Filter 의해 예측된 값을 사용한다. 거리/통합도, 시간/통합도는 거리, 시간을 인지도의 값으로 나눔으로써 계산상의 알고리즘에는 실제거리보다 적은 거리가 입력되게 된다. 그리고 인지도가 높은 링크일수록 더 적은 거리가 입력되게 된다. 따라서 인지도가 높은 경로만을 선택하게 된다.

Integration(통합도) 식의 구성은 다음과 같다.

$$TD_i = \sum_{s=1}^m S \times K_s$$

$$MD_i = \frac{TD_i}{K-1}$$

$$RA_i = 2 \times \frac{MD_i - 1}{K - 2}$$

$$I_i = \frac{1}{RRA} = \frac{RA(D)_i}{RA_i}$$

여기서, TD_i : 공간 i 의 TotalDepth

S : 공간 i 에서 거치게 되는 단계(Step)의 수

m : 공간 i 에서 가장 깊은 공간까지 거치게 되는 단계의 수

K_s : S 단계에서의 공간의 개수

RA_i : 공간 i 의 상대적 비대칭성(Relative Asymmetry)

MD_i : 공간 i 의 평균 깊이(Mean Depth)

K : 분석대상 공간의 총 축선도(Axial line)

RRA : 공간 i 실질적 상대적 비대칭성(Real RA)

$RA_i(D)$: RA_i 와 같은 개수의 공간에 대한 보정치(RA(D-Value))

I_i : 공간 i 통합도(Integration)

2) A* 알고리즘

시간변화적인 네트워크(time-dependent network)에서 방대한 링크간의 조합을 이루고 있는 경로에 대한 경우의 수를 계산하기에는 실제 시스템에서 상당히 많은 시간을 요구하게 된다. 따라서 운전자의 일반적인 행태를 반영하여 탐색영역을 제한함으로써 신뢰성을 유지하면서 효과적으로 경로를 탐색하기 위하여 탐색영역을 제한한다.

탐색 영역 제한은 A* 알고리즘을 적용하면 간단하면서도 기존의 최적경로탐색 알고리즘과 유사한 경로 탐색 결과를 나타낸다는 장점이 있다. A* 알고리즘 적용 과정은 최초 기종점을 잇는 직선의 길이와 X축과의 기울기(각도)를 구한다. 중간위치가 원점이 되도록 모든 점에 대하여 평행 이동시키고, 타원방정식에 의하여 A* 알고리즘의 탐색영역을 제한하여 신뢰성을 유지하는 범위 내에서 효과적으로 경로를 탐색할 수 있다.

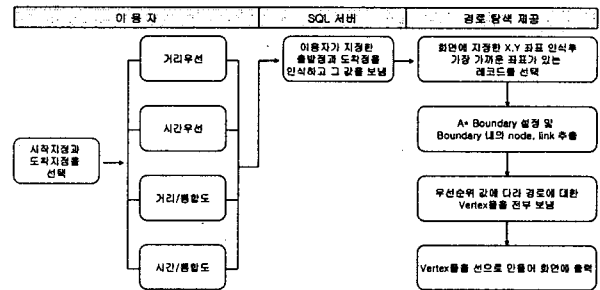
방대한 영역에 대하여 A*Boundary를 생성하면 많은 시간이 소요되어 탐색영역을 제한하도록 설정하였지만, 이것 역시 시작점과 도착점 설정 시마다 동적으로 A*Boundary를 생성함

으로써 Integration값을 도출하는데 매 60초~120초의 시간이 소요된다는 단점이 있었다.

III 프로그램 구현

1. 경로탐색 Code 흐름도

사용자가 시작점과 도착점을 지정하고, 거리, 시간, 거리/통합도, 시간/통합도중의 길을 선택하면 본 연구에 사용된 .shp 파일에서 좌표를 인식하여 경로탐색에 보내준다. 경로탐색은 화면에서 지정한 값을 X, Y 좌표로 인식한 다음, 가장 가까운 좌표가 있는 레코드의 위치를 찾는다. A* Boundary 내에 있는 Node를 읽고, 이들의 링크들을 읽은 다음 Axial Line에 가장 가까운 Node를 찾아서 그에 따른 경로에 대한 Vertex들을 전부 보내면, 보내진 Vertex들로 라인으로 구성하여 화면에 보여 준다.



<그림 7> 경로탐색 분석 과정

2. A* boundary 내의 경로 탐색 알고리즘의 개선

전통적인 Dijkstra Algorithm에서는 L(v)(누적된 링크 weight)를 모든 정점에 대해서 유지한다. 하지만 이와 같은 경우 L(v)의 최소값을 구하는데 전체 정점 수만큼 검색을 해야 한다는 단점이 있다. 따라서 L(v)을 저장하는 방법을 개선하면 효과적으로 성능을 향상할 수 있다.

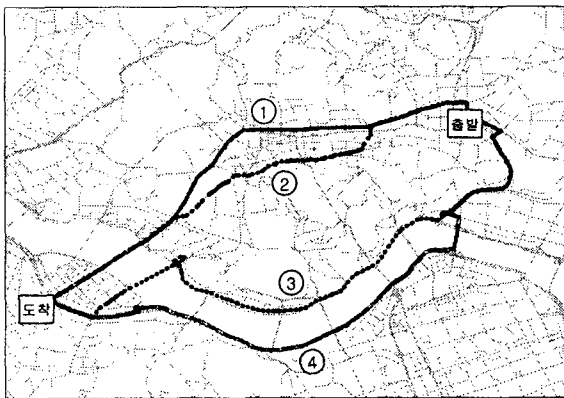
<표 4> 고전적인 Dijkstra 알고리즘과 Bucket구조를 사용한 Dijkstra Algorithm의 비교

구분	실험 1	실험 2	실험 3
Node 개수/ Link 개수	59 / 133	385 / 1025	4300 / 11786
Bucket을 이용한 Dijkstra Algorithm	0.812ms	9.345ms	199.993ms
고전적인 방법의 Dijkstra Algorithm	2.282ms	99.999ms	17999.9993ms
속도 향상 비율	2.8 배	10.7배	90.4 배

본 연구에서는 Bucket을 사용하여 방문했던 노드와 방문하지 않았던 노드에 대해서 따로 관리하도록 프로그램을 구현하여 탐색속도를 개선하였다.

3. 프로그램 구현

운전자 중에는 여러 직업을 가진 운전자가 존재한다. 예를 들어 택시기사나 직장인의 경우는 아무래도 운전하기 편한길 보다는 통행시간이나 통행거리를 중요하게 된다. 그리고 학생이나 기타 통행시간에 구애받지 않는 가정주부의 경우에는 아무래도 운전하기 편한길을 선호하는 경향이 있다. 따라서 본 연구는 사용자가 지정한 우선순위에 따라 여러 경로를 보여줌으로써 사용자가 자신에 맞는 경로를 선택하는데 목적이 있다. 본 연구에서 제공하는 경로는 가장 짧은 거리를 가지는 경로, 시간이 가장 적게 걸리는 경로, 거리를 AxialLine의 통합도로 나누어서 거리와 통합도를 함께 고려한 경로, 시간을 AxialLine의 통합도로 나누어서 시간과 통합도를 함께 고려한 경로통합도가 높은 경로를 제공한다.

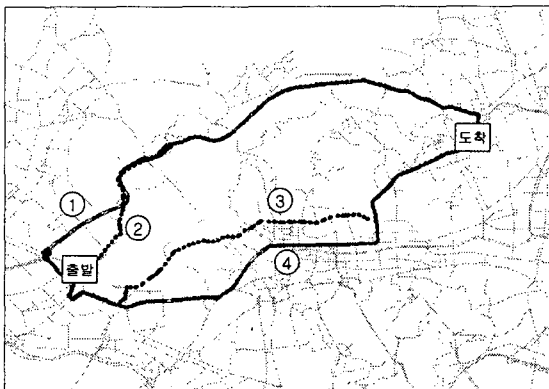


<그림 8> 답십리~영등포

<표 5> 답십리~영등포 최적경로에 대한 결과값

구분	통행 거리	통행시간
1번 경로(인지도_거리)	14,038 m	37분 8초
2번 경로(거리)	13,770 m	34분 10초
3번 경로(시간)	19,968 m	32분 41초
4번 경로(인지도_시간)	20,595 m	33분 21초

1번과 2번 경로는 거리에 대한 경로로서 인지도를 고려한 1번 경로가 2번 경로보다 더 선호하는 것으로 나타났다. 그리고 대부분의 사람이 3번과 4번 경로를 택하였다. 이는 비교적 장거리 통행시 연속류 도로를 선호하기 때문이다.



<그림 9> 경희대~신촌

<표 6> 답십리~영등포 최적경로에 대한 결과값

구분	통행 거리	통행시간
1번 경로(인지도_시간)	16,644 m	22분 9초
2번 경로(시간)	15,041 m	20분 3초
3번 경로(거리)	10,520 m	25분 16초
4번 경로(인지도_거리)	10,641 m	29분 10초

1번, 2번 경로는 시간에 대한 경로이다. 뚜렷한 대안 경로는 나타나지 않았으나 1번 경로가 2번 경로보다 간선도로를 더 많이 이용하는 것으로 나타났다. 이는 1번 경로가 2번 경로에 인지도가 감안됐기 때문이다.

3번, 4번 경로는 거리에 대한 경로로서 4번 경로가 종로를 경유하고 있다. 이는 현재 종로가 인지도가 높기 때문이다. 그에 반해서 3번 경로는 비록 통행거리는 짧으나 구불구불한 도로가 많음을 알 수 있다.

IV 결론 및 향후 연구과제

본 연구는 운전자가 운전할 때 단지 통행시간이 짧거나 통행거리가 짧은길은 선택하기 보다는 자신이 운전하기 편리한길을 선택한다점을 착안하여 연구하게 되었다. 운전하기 편리한 길을 도로를 추출하기 위해서 본 연구에서 적용한 알고리즘은 Space Syntax 이론을 응용하여 인지도가 높은 도로를 판별하였다.

먼저 GIS 파일인 .shp 파일을 이용하기 위해서 .shp 파일을 데이터 파일로 변환이 필요했고 변환된 데이터 파일을 통하여 Space Syntax 이론에 근거한 Axial Map 구축이 필요하였다. Axial Map을 자동으로 구축하기 위해서 수정된 Dijkstra 알고리즘을 이용하였고 Axial Map 구현후에 Dijkstra 알고리즘에 가중치 값을 변환하여 경로탐색 알고리즘을 적용하였다.

본 연구를 나타는 결과값을 평가하기 위해서 약 100명에 설문한 결과 전체적으로 운전하기 편한길, 운전하기 쉬운길을 선호하는 것으로 나타났다. 하지만 설문 10문 중 5문항만이 본 알고리즘의 결과값을 선호하는 것으로 나타났다. 이는 설문자 대부분이 통행시간을 중요시하는 직장인임을 감안하면 어느 정도 타당한 알고리즘이라고 생각된다.

기존의 대안 경로 알고리즘인 K-Shortest 이론의 경우 경로중에서 중복구간이 많이 발생한다는 단점이 있었으나 본 연구에서는 명확하게 대안 경로가 탐색됨을 알 수 있었다.

향후 연구과제로는 통행자의 행태를 더 반영한 최단경로 알고리즘 개발이 더 필요하며 Space Syntax 알고리즘을 교통에 적용시에 단순 network 속성만이 아닌 수요 즉 통행량에 대해서 좀 더 깊게 발전시켜야 될 것이다.

참고문헌

- 1 권희정(2002), User-Equilibrium 통행배정과 O-D 추정에 관한 연구
- 2 김범일(2004) 도로 위계구조를 고려한 동적 최적경로탐색 기법 개발
- 5 김영욱, 김현식(2002) Space Syntax를 활용한 개성공단 개발의 공간적 파급효과, 국토연구원
- 6 이승재외 9명(2001) 동적 최적경로 탐색기술 개발, 과학기술부
- 7 이재규(2001), C로 배우는 알고리즘, 세화
- 8 임강원, 임용택, (2003), "교통망분석론", 서울대학교 출판부
- 9 주우석(2003), C, C++로 배우는 자료구조론, 한빛미디어
10. Adrijana Car and Andrew U.Frank(1993), Hierarchical Street Networks as a Conceptual Model for Efficient Way Finding
11. Roy Thomas(1991) Traffic Assignment Techniques, AVEBURY TECHNICAL
12. Hillier B, Hanson J(1984) The Social Logic of Space, Cambridge University Press
14. Seungjae Lee(2005), An Alternative Shortest Path Algorithm Using Space Syntax Techniques , Working Paper, UCL