

■ 論 文 ■

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 (서울시립대학교 교통공학과) **이 승 재** (서울시립대학교 교통공학과 부교수) **안 우 영** (한국교통연구원 책임연구원)

목 차

- I. 서론
 - 1. 연구의 목적 및 범위
 - 2. 기존 연구 및 문헌 고찰
 - II 다중경로제공 시스템 개발
 - 1. 통행시간 예측
 - 2. Axialmap 구축
 - 3. 경로탐색 알고리즘
 - III. 프로그램 구현
 - 1. 경로탐색 Code 흐름
 - 2. 경로 탐색 알고리즘의 개선
 - 3. 프로그램 구현
 - IV. 결론 및 향후 연구과제
- 참고문헌

Key Words : Kalman Filter, SpaceSyntax, 동적최적경로, 다중경로, GIS

요 약

기존의 최적경로 알고리즘은 통행거리, 통행시간, 통행량 등의 통행값을 통하여 최적경로를 제공하였다. 하지만 이렇게 제시된 최적경로는 사용자의 도로에 대한 인지도를 고려하지 않으므로 자신이 인지하거나 다수의 사용자가 선호하는 경로를 고려하지 못하는 단점이 있었다. 따라서 본 연구에서는 통행거리와 통행시간을 고려하면서 사용자의 인지도를 고려한 최적경로를 개발하는 것이 본 연구의 목적이다. 기존의 통행시간 예측방법에는 ARIMA모형, Kalman Filter모형, 확률과정모형, 신경망모형, 회귀모형 등 여러 가지 방법이 있으나 본 연구에서는 단기 통행시간 예측에 적합한 Kalman Filter 모형을 적용하였다. 인지도를 고려한 최적 경로를 제공하기 위한 기존의 방법은 회전에 대한 가중치를 부여하여 최적경로 탐색시 최소한의 회전을 유도하고 있다. 하지만 회전에 대한 가중치를 주는 방법은 경험적인 방법으로서 만약 신설된 길에 대한 경로 제공, 또는 개량된 길에 대한 경로를 제공할 때 문제점이 나타난다. 본 연구에서는 이 같은 문제점을 해결하고자 공간구조의 속성을 정량적으로 분석하고 평가하는 기법인 Space Syntax 이론을 적용하였다. 운전자들을 대상으로 실시한 설문조사 결과 본 연구에 의한 알고리즘이 기존의 최적 경로보다 더 선호하는 것으로 나타났다.

The object of this paper is to develop the shortest path algorithm. The existing shortest path algorithm models are developed while considering travel time and travel distance. A few problems occur in these shortest path algorithm models, which have paid no regard to cognition of users, such as when user who doesn't have complete information about the trip meets a strange road or when the route searched from the shortest path algorithm model is not commonly used by users in real network. This paper develops a shortest path algorithm model to provide ideal route that many people actually prefer.

In order to provide the ideal shortest path with the consideration of travel time, travel distance and road cognition, travel time is predicted by using Kalman filtering and travel distance is predicted by using GIS attributions. The road cognition is considered by using space data of GIS. Optimal routes provided from this paper are shortest distance path, shortest time path, shortest path considering distance and cognition and shortest path considering time and cognition.

1. 서론

기존의 최적경로 알고리즘은 거리, 시간, 교통 통행량 등의 통행값을 고려하여 구성되었다. 하지만 이렇게 제시된 최적경로는 사용자의 인지도에 대한 고려를 하지 않음으로써 자신이 인지하거나 다수의 사용자가 선호하는 경로를 고려하지 못하는 단점이 있었다. 특히 초행길을 운전하는 경우 더욱 이러한 단점이 크게 작용될 수 있다.

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

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, Kalman Filter 알고리즘의 적용을 통해 정적시간이 아닌 동적 통행시간을 제공하는 것이다.

2. 기존 연구 및 문헌 고찰

GIS를 교통에 적용시킨 부분을 살펴보면 대표적으로 GIS-T로 설명할 수 있다. GIS-T(교통지리정보시스템; Geographic Information System for Transportation)란 지리정보체계(GIS)와 교통관리체계(TIS)를 통합된 체계로 결합시킨 기술을 말하며, GIS를 교통의 목적에 적합하게 변형한 것으로 (Lewis, 1990) 교통에 관련된 공간정보의 분석·관리 (네트워크의 관리와 처리, 네트워크

속성의 갱신과 가시화, 공간분석, 경로분석, 다른 적용분야와의 연계)를 위한 하부구조라고 볼 수 있다 (Lewis and Fletcher, 1992). FHWA(미국 연방 고속도로 관리기관)는 1991년 미국내의 육상교통효율화법(ISTEA: Intermodal Surface Transportation Efficiency Act)에 의거하며 국가 교통기반지정정보체계(National Transportation-based GIS)구축의 첫 단계를 완성하였다. 그리고 미국 통계청(Bureau of Census)에서는 1970년부터 인구센서스를 위해 GBF/DIME 파일이라는 수치지도를 만들어서 사용하였다. 영국에서는 교통부분 지리정보체계의 데이터베이스로 OSCAR파일을 구축하였고 일본에서는 보다 광범위한 RACS(Road Automobile Communication System)를 사용하여 기본적인 표준 데이터베이스를 개발하였다.

기존의 개발된 최적경로 탐색 알고리즘에 대해서는 Larson and Odoni(1981), Papageorgiuo(1991), Thomas(1991), Bertsekas(1992), Magnanti and Orlin(1993) 등의 서적에 잘 서술 되어 있으며, 특히 교통분야에 적용되는 최적경로 탐색에 대해서는 Thomas (1991)의 서적에 이론 및 알고리즘에 대해서 더욱 구체적으로 설명되어 있다. 교통분야에 가장 보편적으로 많이 응용되고 있는 알고리즘 중에는 Dijkstra 알고리즘, Moore 알고리즘, 그리고 D'Esopo 알고리즘 등이 있으며 네트워크가 커질수록 최적경로 탐색 시간이 증가하는 단점을 보완하기위해서 자료구조인 L-deque 또는 Bucket 을 사용하는 알고리즘이 있다.

양방향 탐색 알고리즘은 출발지로부터 탐색을 시도하는 전방 탐색과 목적지로부터의 후방 탐색을 동시에 실시함으로써, 최적해의 탐색속도를 향상시키고자 하는 방법이다. 이 방법은 Dantzig에 최초로 시도되었으며, Nicholson(1996)에 의해 보다 개선된 알고리즘으로 다듬어졌다. 특히, Helgasson은 Dijkstra 알고리즘을 기반으로 두 개의 수형망을 동시에 구축하는 방법이 원래의 Dijkstra 알고리즘보다 약 50%의 탐색 공간 감소효과를 가져올 수 있음을 보여준 바 있다. 또한 최적 경로 알고리즘에는 일반적으로 돌아가지 않고 일정한 영역에 제한적으로 존재한다는 가정하에 탐색 영역을 제한하는 A*알고리즘이 있다.

구간 통행시간 예측을 위하여 지금까지 사용한 대표적인 방법은 통계적 기법을 이용하는 것이다. 이들 기법중에는 다중회귀분석, 시계열분석, Kalman Filter,

신경망 이론이 있으며 이승재(2004)의 연구에 의하면 단기통행시간 예측으로서는 Kalman Filter가 가장 적합한 알고리즘으로 분석되었다.

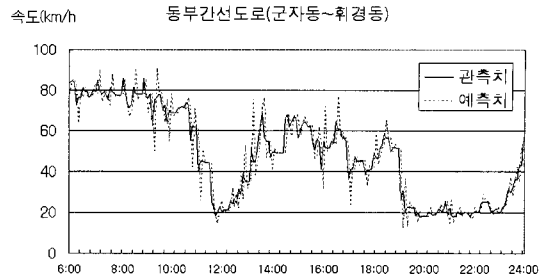
Space Syntax 방법론은 공간구조를 분석하여 각 공간의 속성을 정량적으로 제시하는 이론이자 이를 토대로 개발된 일련의 컴퓨터 프로그램을 칭한다. 본 방법론은 1980,1990년대에 걸쳐 영국 런던대학교의 Hillier 교수 연구팀이 개발하였다. 위 이론은 공간구조상 중요도를 분석대상지역의 전체 공간에서의 접근성에 의하여 계산한다. 즉, 분석대상 범위내의 모든 공간이 기점이자 종점이 되는 가정아래 각 공간의 접근성을 분석하게 된다. 이 접근성을 Space Syntax에서는 전체 공간을 통합하여 주는 통합성 혹은 공간구조상의 위계성을 의미하는 통합성(Integration: 인티그레이션) 이라 정의한다. 따라서 Space Syntax 분석 결과에 의한 통합성이 큰 공간은 다른 모든 공간으로부터의 접근성이 양호하다는 것을 의미한다.

II 다중경로 시스템 개발

1. 통행시간 예측

Kalman Filter는 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 예측 방법 중에서 Kalman Filter에 의한 방법이 가장 뛰어난 것으로 나타났다. 따라서 본 연구에서는 최적경로중의 하나로 Kalman Filter에 의해서 예측된 통행시간을 기반으로 최적경로를 제공하였다.

본 연구의 공간적 범위 안에는 총 24,000개 링크중 약 15,000개의 링크에 대해서 통행시간 예측이 가능하기에 약 15,000링크에 대해서 통행시간을 예측 하였고 나머지 9,000개의 링크에 대해서는 자유속도를 적용하여 통행시간을 설정하였다. 이는 현재 9,000의 링크는 위계가 낮은 도로이기 때문에 자유속도를 통하여 통행시간을 설정하여도 무방하기 때문이다.



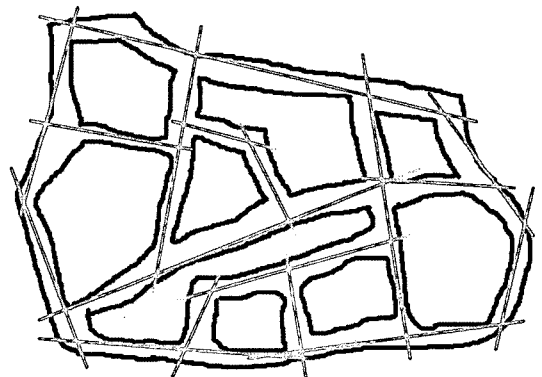
〈그림 1〉 Kalman Filter에 의한 통행시간 예측

2. Axialmap 구축

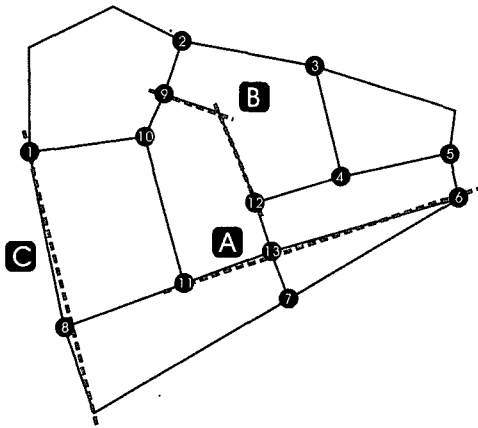
기존의 Space Syntax 분석 프로그램은 Axial Map을 입력하면 분석이 가능하지만 사용자가 Network에 근거한 Axial Map을 일일이 수작업으로 작성해야 하는 단점이 있었다. 또한 대규모 Network에 대해 Axial Map을 작성하는 것은 많은 시간이 소요되고, 분석하는 개인의 주관에 개입되기 쉬운 단점도 있었다. 따라서 Network에 근거한 Axial Map을 컴퓨터가 자동으로 구성하도록 하는 방법을 고안하고 개발하여 본 연구에 적용하였다.

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

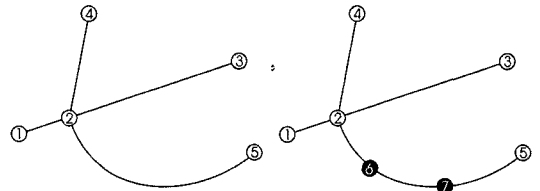
Space Syntax의 Axial Line과 교통 Network는 서로 대응되지 않는다. 실제의 도로망과 Axial Map은 양쪽 모두 수학적인 Graph로 모델을 추상화를 할 수 있으나 이 두 개의 모델에서 node와 link가 의미하는 바는 서로 다르다.



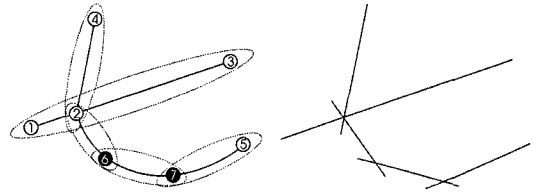
〈그림 2〉 수작업으로 그려진 Axial Map



〈그림 3〉 GIS교통망의 Graph표현



(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

〈표 1〉 Axial Line과 도로의 대응관계

| 유형 | 설명 | 예제 |
|----|-----------------------------------|--|
| A | 단일 도로가 다수의 Axial Line으로 구성되는 경우 | 16번과 13번 노드를 잇는 링크는 2개의 Axial Line임 |
| B | 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 |
| C | 인접한 링크의 일부만이 서로 모여 Axial Line을 이룸 | 1번과 2번을 잇는 링크는 2번과 4번을 잇는 링크의 일부분과 함께 하나의 Axial Line을 구성함 |

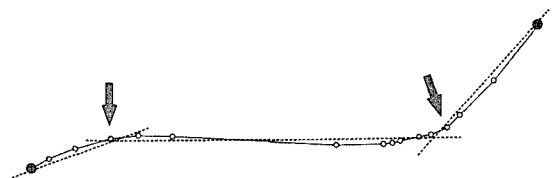
2) Axial Map을 자동으로 구성하기 위한 Algorithm 개발

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 개발 하였다.

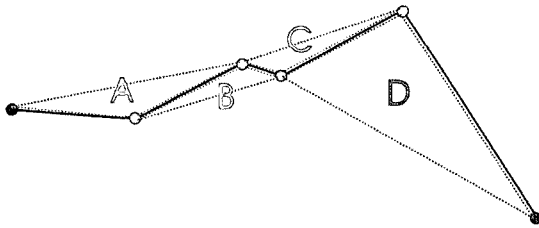
- 도로 Network에서 각 링크가 직선인지 곡선인지 판단
- 곡선링크일 경우 적절한 더미 노드를 추가하여 다수의 링크로 분할
- 추가된 Node를 포함하여, 기울기가 비슷하고 연결된 링크 집합을 구함
- 구해진 각각의 집합에 고유의 Axial Line ID 를 부여
- 각각의 Axial Line ID(Space Syntax Index)에 근거하여 상호 접근성을 측정하여 Axial Map을 산출

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 실제로 작업을 할 경우, 일관성 없는 결과를 낳게 된다. 곡선 링크를 몇 개의 Axial Line으로 볼 것인가 하는 것은 주관적인 판단에 의해 변할 수 있으므로, 프로그램에 적당한 파라메타를 입력함으로써 동일한 기준으로 Axial Line의 개수를 객관적으로 판단 할 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



〈그림 5〉 기울기의 변화 측정을 이용한 Axial line 판단

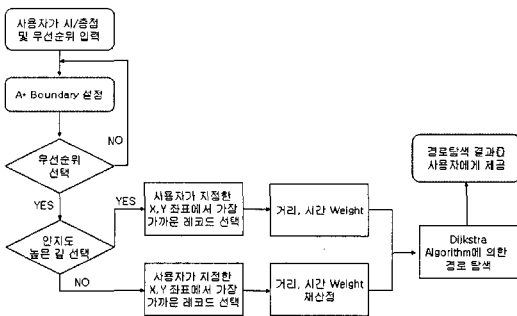


(그림 6) 면적을 이용한 Axial line 판단

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 최대 길이의 Axial Line을 산출할 수 있다.

3. 경로탐색 알고리즘

사용자가 출발점과 도착점을 지정하고 가장 짧은 거리를 가지는 경로, 시간이 가장 적게 걸리는 경로, 거리를 AxialLine의 통합도⁹⁾로 나누어서 통합도가 높은 경로를 사용자가 선택하고 출발점과 도착점을 입력하면 경로탐색시스템은 사용자가 지정한 X, Y좌표에서 가장 가까운 레코드를 선택하고, 결절점(Node)을 검색해서 최단경로를 찾아준다.



(그림 7) 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 연

결된 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.

본 연구에서는 운전자가 통행을 시작할 때의 교통상황을 기반으로 하지 않고, 운전자가 실제로 링크에 도달했을 경우 그 시간대의 링크의 통행시간의 예측하여 이를 바탕으로 통행시간 정보를 제공하였다. 위와 같은 동적인 통행시간은 다음의 수식으로 나타 낼 수 있다.

$$C_p^n(t) = C_{a1}(t) + C_{a2}(t) + \dots + C_{am}(t)$$

$$C_p^d(t) = C_{a1}(t) + C_{a2}(t + C_{a1}(t)) + \dots + C_{am}(t + C_{a1}(t) + C_{a2}(t + C_{a1}(t)) + \dots + C_{am-1}(t + C_{a1}(t) + \dots + C_{am-2}(t)))$$

여기서, $C_p^n(t)$: 시간 t 에 출발하여 경로 P 를 이용할 때의 정적경로 통행시간

$C_p^d(t)$: 시간 t 에 출발하여 경로 P 를 이용할 때의 동적경로 통행시간

$C_i(t)$: 시간 t 에 링크 a 통행시간, $a1, a2, \dots, am \in p$

경로로 제공되는 최적 거리 경로, 최적 시간 경로, 인지하기 쉬운 경로 등에 따라 알고리즘에 사용되는 통행 비용이 다르게 적용된다. 최적 거리 경로의 경우는 거리를 비용으로 사용하여 거리가 최소화 되는 경로를 제공하며, 최적시간 경로의 경우 여러 단통행시간 예측기법중에서 정확성이 가장 우수한 Kalman Filter 방법을 적용하여 통행시간이 최소화되는 경로를 제공한다. 인지하기 쉬운 경로는 Space Syntax의 통합도값을 이용하여 최적경로를 제공한다. 단 통행시간경로의 경우 정적통행시간이 아닌 동적통행시간을 사용하며 인지도를 고려한 통행시간 최적 경로의 경우도 동적통행시간을 사용한다.

9) 임의의 공간에서 다른 공간까지 접근하기 위한 상대적인 깊이와 관련한 변수로, 통합도가 높은 값을 가지면 가질수록 임의의 공간으로부터 그 공간까지로의 이동 단계가 적어지게 되어 접근하기 쉬운 것을 의미한다.

〈표 2〉 본 연구에 제공된 다중경로

| 다중경로 | 경로 제공 방법 |
|--------|------------------|
| 거리 | node i까지의 거리 |
| 시간 | node i까지의 통행시간 |
| 거리/통합도 | node i까지의 거리+통합도 |
| 시간/통합도 | node i까지의 시간+통합도 |

거리는 경로가 가지는 길이의 총합을 말하며, 시간은 Kalman Filter 의해 예측된 시간을 바탕으로 한 동적 통행시간을 말한다. 거리/통합도, 시간/통합도는 거리, 시간을 인지도의 값으로 나눔으로써 계산상의 알고리즘에는 실제거리보다 적은 거리가 입력되게 되고 인지도가 높은 링크일수록 더 적은 거리가 입력되게 된다. 따라서 인지도가 높은 경로를 선택하게 된다.

Integration(통합도) 식의 구성은 다음과 같다.

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

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

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

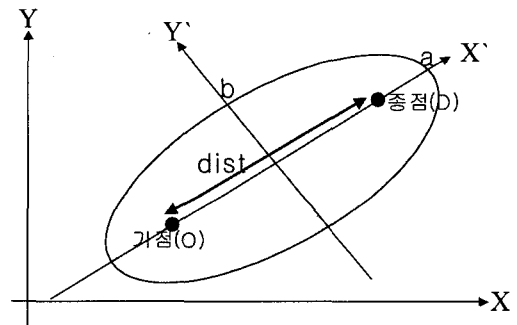
$$I_i = \frac{1}{RRA} = \frac{RA(D)_i}{RA_i}$$

- 여기서, TD_i : 공간 i 의 Total Depth
- 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 비대칭 (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 기증점을 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.

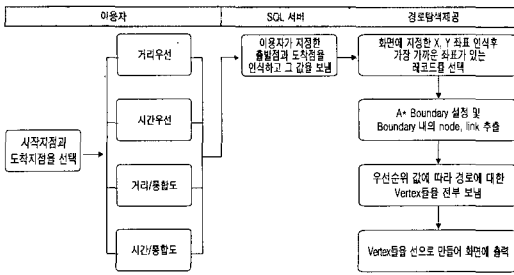


〈그림 8〉 A* 탐색영역 설정

III 프로그램 구현

1. 경로탐색 Code 흐름도

사용자가 시작점과 도착점을 지정하고, 거리, 시간, 거리/통합도, 시간×통합도중의 길을 선택하면 본 연구에 사용된 GIS 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 보여준다.



〈그림 9〉 경로탐색 분석 과정

2. 경로탐색 알고리즘의 개선

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

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

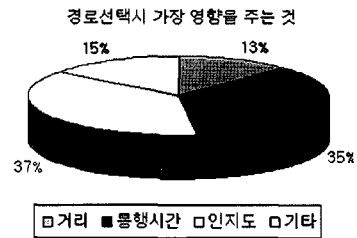
〈표 3〉 고전적인 Dijkstra 알고리즘과 Bucket구조를 사용한 Dijkstra Algorithm의 비교

| 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 | 17999.999ms |
| 속도 향상 비율 | 2.8 배 | 10.7배 | 90.4 배 |

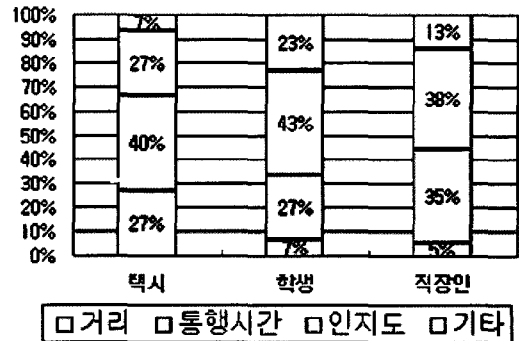
3. 프로그램 구현

운전자 중에는 여러 직업을 가진 운전자가 존재한다. 예를 들어 택시기사나 직장인의 경우는 아무래도 운전하기 편한길 보다는 통행시간이나 통행거리를 중요시하게 된다. 그리고 학생이나 기타 통행시간에 구애받지 않는 전업주부의 경우에는 아무래도 운전하기 편한 길을 선호하는 경향이 있다. 따라서 본 연구는 사용자가 지정한 우선순위에 따라 여러 경로를 보여줌으로써 사용자가 자신에 맞는 경로를 선택하는데 목적이 있다

본 연구에서는 이를 평가하기 위해서 100부의 설문조사를 수행하였으며 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 경로에 대해서 설문하였다.



직업별 분포



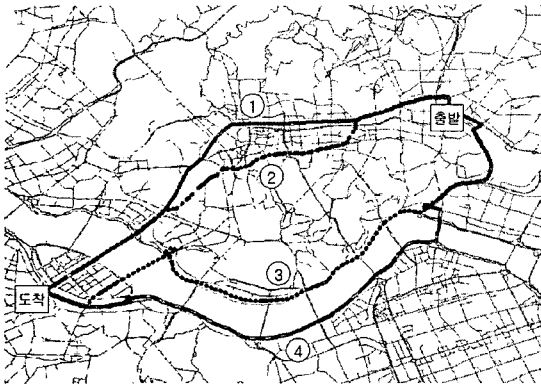
〈그림 10〉 경로선택시 가장 영향을 주는 요소

경로 선택시 대부분의 사람들이 통행시간과 경로에 대한 인지도를 중요시 하는 것으로 나타났다.

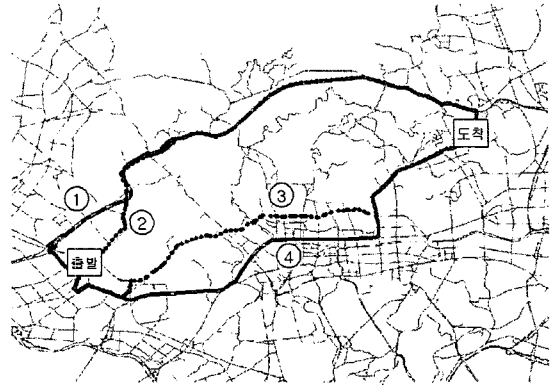
직업별 분포를 살펴보면 택시운전자의 경우 상황에 따라 경로를 선택하는 것으로 나타났으며 학생의 경우 인지도가 높은 길을 선택하는 것으로 나타났다. 또한 직장인의 경우 대부분 출근시에는 통행시간을 중시하며 퇴근시에는 인지도가 높은 길을 선호하는 것으로 나타났다.

기타 의견으로는 “안전한길”, “운전하기 편한길”, “지체가 없는 길” 등이 있었다.

1번과 2번 경로는 거리에 대한 경로로서 인지도를 고려한 1번 경로가 2번 경로보다 더 선호하는 것으로 나타났다. 이는 현재 1번 경로가 2번 경로보다 인지도가 좋은 그리고 대부분의 사람이 3번과 4번 경로를 택하였는데 이는 통행거리보다는 통행시간의 요소가 많이



〈그림 11〉 답십리~영등포



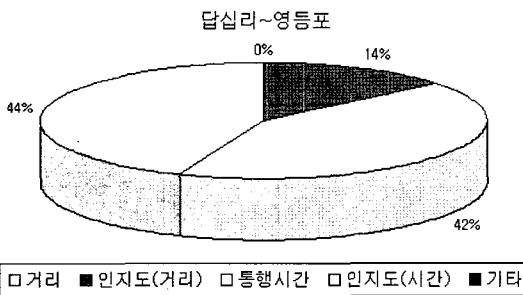
〈그림 13〉 경희대~신촌

〈표 4〉 답십리~영등포 최적경로에 대한 결과값

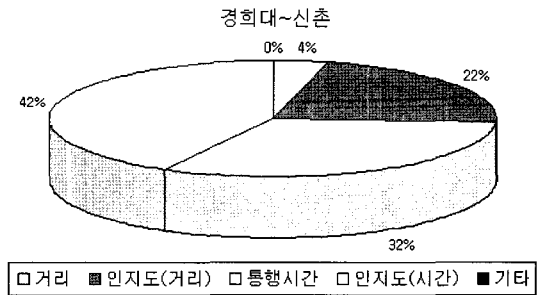
| 구 분 | 통행 거리 | 통행시간 |
|---------------|----------|---------|
| 1번 경로(인지도_거리) | 14,038 m | 37분 8초 |
| 2번 경로(거리) | 13,770 m | 34분 10초 |
| 3번 경로(시간) | 19,968 m | 32분 41초 |
| 4번 경로(인지도_시간) | 20,595 m | 33분 21초 |

〈표 5〉 경희대~신촌 최적경로에 대한 결과값

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



〈그림 12〉 답십리~영등포 구간에 대한 선호도



〈그림 14〉 경희대~신촌 구간에 대한 선호도

반영됨을 알 수 있다. 그리고 비교적 장거리 통행시 연속 도로를 선호하기 때문이다.

〈그림 13〉의 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 인지성이 높기 때문이다. 그에 반해서 3번 경로는 비록 통행거리가 짧으나 구불구불한 도로가 많음을 알 수 있다.

IV. 결론 및 향후 연구과제

본 연구는 1,000개 노드 이상의 현 도심교통 네트워크에서 사용자에게 인지도가 높은 최적경로제공을 목표로 하였다. 연구의 수행방법은 최근 대두되는 Space Syntax이론과 Kalman Filter 이론을 기반으로 하였다.

우선 GIS를 기반으로 한 교통정보제공 시스템을 구축하기 위해서 NGIS 구축 사업의 일환으로 만들어진 국가교통 주제도와 현재 Car-Navigation 회사에서 사용하고 있는 GIS 데이터를 경로탐색 알고리즘에 적합한 네트워크 체계로 변환하는 작업이 우선적으로 수행되었다.

동적 최단 경로를 제공하기 위해서는 통행시간 예측이 필요하다고 판단되었고 Kalman Filter 알고리즘을 적용하여 5분 주기로 장래통행시간을 예측 하였다. 현

제 서울시내 모든 도로에 대한 통행속도 자료수집이 어렵기 때문에 수집되지 않는 도로에 대해서는 자유속도를 적용하였다. 이는 현재 대부분 주요한 도로는 자료수집이 가능하기 때문에 큰 영향이 없으며 실제로 Car- Navigation 회사에서도 자료수집이 불가능한 도로에는 자유속도를 적용하여 경로를 제공하고 있기 때문이다.

인지도가 높은 길을 제공하기 위해서 수정된 네트워크를 바탕으로 Space Syntax 이론을 적용한 Axial Map을 구축이 필요하였다. Map 구축시 기존에는 수작업으로 진행하여 많은 시간과 노력이 필요했고 사람의 주관이 개입될 여지가 있었으나 프로그래밍 알고리즘을 통한 Map 구축으로 기존의 단점을 해결할 수 있었다.

기존의 교차로별 노드체계에서 기존 알고리즘의 사용은 알고리즘의 수행 속도측면에서 너무 많은 단점을 갖고 있었기에 본 연구에서는 알고리즘의 변형을 통하여 이러한 단점들을 극복할 수 있었다. 경로탐색 알고리즘은 Dijkstra 알고리즘과 자료구조 알고리즘인 Bucket을 사용하였으며 탐색영역 제한기법 알고리즘인 A* 알고리즘을 적용하였다. 탐색영역 제한기법은 타원방정식을 기본으로 하였고, 영역제한에 대한 파라메타 λ 를 사용하여 유동성을 내제할 수 있도록 하였다.

시스템 구축은 Microsoft Visual Studio 6.1(Microsoft Visual Studio 6 Service Pack 5)를 이용하였고 GIS데이터의 Vertex정보를 분석하기 위해 SQL서버 200을 사용하였다.

마지막으로 본 연구를 평가하기 위해서 설문조사를 수행하였다. 설문조사는 "경로탐색시 가장 중요한 요소"를 첫 문항으로, 각각의 경로에 대하여 선택하는 방식으로 구성되었다. 또한 각각의 직업별로 선호하는 경로가 다르기 때문에 설문대상자의 경우 택시기사, 학생, 직장인으로 세분화 하여 조사하였다. 분석결과 대부분의 사람들이 본 연구를 통해 개발된 최적경로알고리즘을 기존에 제공했던 최단거리 최적경로, 최단시간 최적경로보다 더 선호하는 것으로 나타났다.

본 연구의 문제점은 현재 적용된 Space Syntax이론이 방향성을 고려하지 못한다는 것이다. 이는 Axial Map을 구축할 때 통행수요 보다는 단지 네트워크에 나타는 속성치만을 대상으로 하기 때문에 나타나는 문제점이라고 생각된다. 그리고 현재 Axial Map 구축시 교통네트워크에 맞는 Calibration을 어떤 방법에 의해서 할 것인가에 대한 문제가 존재한다.

또 하나의 문제점은 인지도 높은길에 대한 정의가 명확하지 않다는 것이다. 다시 말하면 인지도라는 개념이 과연 어떠한 계층까지 적용 되는가에 되는 문제이다. 예를 들어 어떠한 개인이 초행길이라면 본 알고리즘이 타당하지만 만약 운전은 자주하거나 자신만의 익숙한 길이 있다면 본 알고리즘은 적용하기 어려울 것이다.

따라서 이를 개선하기 위해서는 통행자의 행태를 반영한 최적경로 알고리즘 개발이 필요하며 Space Syntax 알고리즘을 교통에 적용시에 단순 네트워크 속성만이 아닌 수요 즉 통행량에 대해서 좀 더 깊게 발전시켜야 될 것이다.

참고문헌

1. 김익기(1998), ATIS를 위한 수정형 덩굴망 최단 경로 탐색 알고리즘의 개발, 대한교통학회지, 제 16권 제2호, 대한교통학회, pp.157~167.
2. 김영옥 · 한상진 · 신행우(2005), "보행네트워크 분석을 통한 보행량 예측 방법 연구 -Space Syntax 기법을 활용하여-", 제48회 학술발표회, 대한교통학회, pp.365~374.
3. 김범일(2004), "도로 위계구조를 고려한 동적 최적경로탐색 기법 개발", 석사논문.
4. 이경로(2005), "GIS적용을 위한 Axial Map 자동생성 알고리즘 개발", 국토학회.
5. 이승재 · 이영인(2004), "Development of Dynamic Route Guidance Alogrithm Usign Multi-Step ahead Link Travel Time Prediction", BiNs, 서울시립대학교.
6. 임강원 · 임용택(2003), "교통망분석론", 서울대학교 출판부.
7. 전철민(2005), "Assessing Public Transportation Accessibility Based on Topological Structure", CUPUM 05, UCL.
8. 최기주 · 신치현(1998), "GPS와 GIS를 이용한 링크통행시간 예측기법", 대한교통학회지, 제16권 제 2호, 대한교통학회, pp.197~207.
9. 최찬영(1999), "Kalman Filter를 이용한 도시고속도로의 구간속도 예측에 관한 연구", 석사논문.
10. Ahmed. M, Cook. A(1979), "Analysis of Freeway Traffic Time Series Data by Using Box-Jenkins Techniques", Transportation Research Board.

- No. 722, pp.1~9
11. Batty M. and Rana S.(2004), "The automatic definition and generation of axial lines and axial maps", *Environment and Planning B: Planning and Design*, Vol 31, pp.615~640.
 12. Carlos F. Daganzo(1998), "Reversity of the time-dependent shortest path problem"
 13. Dalton N.(2003), "Storing directionality in axial lines using complex node depths", 4th International Space Syntax Symposium London 2003
 14. Fu. L(2005), "Heuristic shortest path algorithms for transportation applications: State of the art", *Computers & Operations research*.
 15. Hiller, B. and Hansom, J.(1984). "The Social Logic of Space", Cambridge University Press.
 16. Roy Thomas(1991), "Traffic Assignment Techniques", Avebury Technical.
 17. Steven I-Jy Chien(2003), "ynamic Travel Time Prediction with Real-Time and Historic Data", *Journal of Transportation Engineering*.
 18. Van Arem. B(1997), "Recent Advances in the Field of Short-term Traffic Forecasting", *International Journal of Forecasting* Vol,12, pp.1~12.

✉ 주 작 성 자 : 류승규

✉ 교 신 저 자 : 이승재

✉ 논문투고일 : 2005. 10. 29

논문심사일 : 2005. 11. 30 (1차)

2005. 12. 15 (2차)

심사판정일 : 2005. 12. 15

✉ 반론접수기한 : 2006. 4. 30