

스마트폰에서 보행자 길안내 서비스를 위한 관심지점 기반 도로 네트워크 모델링 연구

정원일^{1*}, 김상기²

¹호서대학교 정보보호학과, ²인하대학교 컴퓨터정보공학과

A Study on Road Network Modeling over POI for Pedestrian Navigation Services in Smart Phones

Weonil Chung^{1*} and Sang-Ki Kim²

¹Dept. of Information Security Engineering, Hoseo University

²Dept. of Computer and Information Engineering, Inha University

요 약 최근 GPS 기능을 내장한 스마트폰의 보급이 확대됨에 따라 위치를 기반으로 하는 다양한 응용 서비스들이 제공되고 있다. 기존의 보행자 길안내 서비스는 차량 길안내 서비스를 확장하여 경로 질의를 처리하고 있으나, 인도에 대한 적절한 경로 안내가 어려울 뿐 아니라 차량을 위한 도로 네트워크 저장구조는 관심지점 위주의 경로 질의 처리에 문제를 야기한다. 또한, 저장 장치의 입출력 빈도수를 증가시켜 스마트 폰의 전력 소모를 가중시키게 된다. 본 논문에서는 보행자 경로의 정확한 표현을 위해 보행자 도로 네트워크 모델을 정의하고 저장 장치의 입출력 빈도수를 줄이기 위한 관심지점 기반 보행자 도로 네트워크 저장 구조를 제안한다. 제안 기법은 보행자 도로의 특성과 요구사항에 따른 관심지점 기반의 정확한 경로 탐색을 지원한다. 또한, 관심지점을 고려한 데이터 구조를 제공하여 경로 탐색 질의 처리 비용을 감소시켜 스마트폰의 전력 소모를 줄일 수 있다.

Abstract Recently, the wide spread popularity of smart phones causes the advent of various mobile applications base on the location information. Since previous pedestrian navigations are applied by extending car navigations, these are not only difficult to provide the appropriate route information, but also raise limitations in the efficient query processing by data structures of car road networks. In addition, these increase the power consumption caused by the growth of I/O frequency. In this paper, we propose a pedestrian road network model for the accurate route information and a storage structure for the pedestrian road network based on POI to reduce the I/O frequency. The proposed method enables efficient route searches over POI reflecting the characteristics and requirements of pedestrian roads. Also, a reduction of query processing costs for the route searching by a data structure considered with POI can save the power consumption more than previous approaches.

Key Words : Pedestrian Navigation, Road Network Model, Smart Phone

1. 서론

최근 GPS를 이용한 스마트폰에서 위치 정보와 지리정보 기술을 활용한 다양한 위치 기반 서비스들이 제공되고 있다[1,2]. 특히, 보행자 길안내 서비스는 기존의 차량 도로 네트워크를 이용하여 스마트폰의 제한된 배터리 용

량을 고려한 경로 탐색 질의 처리에 대한 연구가 진행되고 있다[3-7].

모바일 단말기에서 효율적인 위치기반 응용 서비스를 위해 지도상에 객체의 이동 경로를 나타낼 수 있는 도로 네트워크의 표현 방법으로 제안된 Jensen의 모델[3]은 차량 도로에 대한 정확한 표현이 가능해지고 차량 도로에

*교신저자 : 정원일(wnchung@hoseo.edu)

접수일 10년 09월 08일

수정일 (1차 10년 11월 23일, 2차 10년 12월 13일)

게재확정일 11년 01월 13일

대한 렌더링 속도를 높일 수 있는 방법을 제안하고 있다. 이러한 차량 도로 네트워크 모델을 이용하여 경로 질의를 처리하기 위한 저장 기법은 CCAM[8]이 있다. CCAM은 도로 네트워크에서 일정한 개수로 노드를 그룹 단위로 서브 네트워크를 생성하여 각 서브 네트워크 내에 존재하는 노드는 인접 리스트를 이용해 같은 데이터 페이지에 저장하고, 빠른 데이터 접근을 위한 색인을 제공하고 있다. 그러나 기존의 차량 도로 네트워크를 확장하여 제공되는 보행자 길안내 서비스는 골목길, 횡단보도, 육교와 같은 보행자 도로의 정확한 경로의 표현이 어렵다. 또한, 보행자 경로 탐색을 위한 관심지점 기반의 OSR 질의[9], TPR 질의[10], MRPSR 질의[11] 등을 처리하기 위한 도로 네트워크 구성과 인덱스가 구축되지 않아 경로 질의 처리 성능의 저하가 발생한다. 이는 디스크 입출력 빈도의 증가 및 탐색 비용의 증대를 야기하여 스마트폰의 배터리 전력 소모량이 증가시키게 된다. 따라서 스마트폰 기반의 보행자 길안내 서비스를 위해서는 보행자 경로를 위한 도로 네트워크 모델과 함께 이를 기반으로 하는 관심지점 기반의 경로 질의 처리 및 디스크 입출력 비용을 최소화할 수 있는 저장 기법이 요구된다[12-15].

본 논문에서는 효과적인 보행자 길안내 서비스를 제공하기 위한 보행자 도로 네트워크 모델과 관심지점을 기반으로 보행자 도로 네트워크 모델을 이용한 질의 처리 및 저장 기법을 제안한다. 제안 모델은 실제 보행자가 이동 가능한 경로 정보를 표현하므로 보행자 도로의 정확한 표현을 가능하게 하며, 관심지점 기반의 보행자 경로 탐색 질의 처리에 있어 디스크 입출력 비용을 감소시켜 스마트폰의 전력 소모를 줄일 수 있다.

2. 관련 연구

2.1 차량 도로 네트워크 모델

차량을 위한 경로 질의 처리를 위해서는 대량의 공간 데이터에 대한 디스크 접근이 요구되므로, 경로 질의 처리 속도 향상을 위해 공간 데이터와 융합된 차량 도로 네트워크 모델에 대한 연구가 진행되었다[12,16].

Jensen의 연구에서 차량을 위한 도로 네트워크를 표현하기 위해 Kilometer-Post, Link-Node, Geographical, Segment의 네 가지 관점으로 모델을 정의하였다. Kilometer-Post 표현은 행정 구역으로 나누어진 도로의 번호와 길이를 기준으로 도로 네트워크를 표현하고 도로의 실제 지리적인 위치를 디스크에 저장하게 되므로, 도로 위에 존재하는 객체의 위치 파악이 쉽고 실제 지도상에 빠르게 도로

가 렌더링 될 수 있는 장점을 갖고 있다. Link-Node 표현은 교차로와 같은 특수한 지점을 노드로 정하고 두 노드를 연결하여 링크를 표현한다. 노드와 노드를 연결한 링크는 가중치 정보를 함께 갖고 있어 경로 질의 처리 시 링크의 가중치 정보를 이용해 최단 거리 경로에 대한 신속한 질의 처리가 가능하다. Geographical 표현은 실제 지리 좌표를 이용하여 도로의 위치를 표현하므로 정확한 위치를 참조할 수 있기 때문에 이동 객체의 지리 좌표에 대한 정보를 얻을 수 있다. Segment 표현은 도로의 교차점을 이용하여 도로를 나누기 때문에 네트워크 위상과 도로의 집합을 표현할 수 있다. 이와 같은 네 가지의 표현 방식의 관계를 표현함으로써 차량 도로에 대한 정확한 표현이 가능해지고 차량 도로에 대한 렌더링 속도를 높일 수 있다. 그러나 보행자 길안내의 경로 질의에 대한 고려를 하지 않아 상점의 개점이나 폐점 시간과 같은 관심지점의 시간 속성을 고려하지 않고 짧은 거리 위주의 경로를 선택하도록 모델이 정의되었다. 이에 보행자 경로 질의 처리 시 디스크의 입출력 비용이 크고 보행자 도로의 정확한 경로 안내가 어려운 단점이 있다.

2.2 차량 도로 네트워크 저장 기법

차량 도로 네트워크의 저장 방식은 노드의 개수를 서브 네트워크로 구성하고, 서브 네트워크 내에 있는 노드들을 인접 리스트로 구성하여 디스크 데이터 페이지에 저장한다. 인접 리스트들을 최대한 같은 데이터 페이지에 저장하여 디스크의 입출력 횟수를 최소화하기 위한 대표적인 연구로 CCAM이 있다.

CCAM에서 인접 리스트의 자료 구조는 노드 데이터, 선행 노드 리스트와 후행 노드 리스트로 구성된다. 노드 데이터에는 노드의 고유 ID가 저장되어 있으며 선행 노드 리스트는 현재 노드에서 다른 노드로 진출하는 노드를 가리키며 후행 노드 리스트는 다른 노드에서 현재 노드로 진입을 뜻한다. 서브 네트워크는 하나의 데이터 페이지에 저장될 수 있는 인접 리스트의 개수만큼 그룹화하여 생성된다. CCAM은 클러스터링을 통해 인접한 노드들을 같은 데이터 페이지에 저장함으로써 현재 위치에서 목적지까지 안내해주는 경로 질의의 최단 경로 연산 속도가 빠르고 일차원의 데이터만을 디스크에 저장하여 디스크의 공간을 절약할 수 있는 장점이 있다. 그러나 관심지점 기반의 경로 질의 처리 시 관심지점 기반의 인덱스가 구축되어 있지 않고 사용자가 정한 규칙의 개수가 많은 경우 인덱스의 재탐색 시간과 디스크의 입출력 비용이 증가하게 되는 단점이 있다.

3. 보행자 도로 네트워크 모델링

3.1 보행자 도로 네트워크 모델

3.1.1 네트워크 구조

제안 보행자 도로 네트워크 모델은 기존의 차량 도로 네트워크 모델에서 지원하지 않는 대중교통노선, 횡단보도, 육교, 지하도록, 길가장자리구역, 좁은 골목길 등의 경로를 지원한다.

보행자 도로는 차량 전용 도로 외의 나머지 도로로 규정하며, 보행자만 진입할 수 있는 골목길, 산책로 등의 보행자 전용도로, 차량과 보행자 모두 진입이 가능한 횡단보도 등의 공유 도로, 보행자가 차량을 이용해 차량 도로에 진입할 수 있는 교통로로 구분한다. 이와 같은 보행자 도로는 차량 도로보다 많은 도로 타입을 정의하고 표현할 수 있어야 하며, 관심지점 기반의 경로 질의 처리 속도 향상을 위해 관심지점을 기준으로 한 도로 네트워크 표현 방법이 요구된다.

보행자 도로 네트워크는 임의의 관심지점들의 집합 PV(POI Vertex)와 PV를 연결한 간선 E가 주어질 때, 보행자 위치 PPos(G)는 이동 가능한 경로 위에 존재하는 보행자의 위치를 의미한다.

보행자 도로는 PM(Pedestrian Mall), PCR(Pedestrian with Car Road), PTR(Public Traffic Road)로 정의하며, 아래 정의에서 POI 정보(POI_CATEGORY, POI_NAME, LOCATION)는 POI의 범주, 이름 및 위치를 나타낸다.

[정의 1] PM

임의의 p_1, p_2 가 관심지점의 집합에 속하고 p_1 과 p_2 를 연결한 링크 l 이 간선 E 의 집합에 존재할 때, $PM = \{pm \mid pm \in PM, p_1, p_2 \in PT \wedge l \in RC\}$, $Dom(PM) = \{(PM_ID, PM_Len, E_POI) \mid PM_ID \in INT, PM_Len \in INT, E_POI \in POI_INFO\}$

PM은 차량의 접근이 통제된 보행자만이 이동할 수 있는 도로로, PM_ID는 도로의 고유 ID, PM_Len은 도로의 길이, E_POI는 목적지 POI에 해당한다.

[정의 2] PCR

임의의 p_1, p_2 가 관심지점의 집합에 속하고 p_1 과 p_2 를 연결한 링크 l 이 간선 E 의 집합에 존재할 때, $PCR = \{pcr \mid pcr \in PCR, p_1, p_2 \in PT \wedge l \in RC\}$, $Dom(PCR) = \{(PCR_ID, PCR_Len, E_POI) \mid PCR_ID \in INT, PCR_Len \in INT, E_POI \in POI_INFO\}$

PCR은 횡단보도와 같이 보행자의 접근이 가능한 도로로, PCR_ID는 도로의 ID, PCR_Len은 도로의 길이, E_POI는 목적지 POI를 의미한다.

[정의 3] PTR

임의의 p_1, p_2 가 관심지점의 집합에 속하고 p_1 과 p_2 를 연결한 링크 l 이 간선 E 의 집합에 존재할 때, $PTR = \{ptr \mid ptr \in PTR, p_1, p_2 \in PT \wedge l \in RC\}$, $Dom(PTR) = \{(PTR_ID, PTR_Len, E_POI, PT_INFO) \mid PTR_ID \in INT, PTR_Len \in INT, E_POI \in POI_INFO, PT_INFO \in TRAFFIC_INFO\}$

PTR은 대중 교통으로 보행자가 이동할 수 있는 도로로, PTR_ID는 도로의 고유 ID, PTR_Len은 보행자 도로의 길이, E_POI는 목적지 관심지점을 나타내며, PT_INFO는 버스 및 전철에 대한 정보로 실시간 대중교통의 위치를 포함한다.

3.1.2 네트워크 연산

보행자 도로네트워크 연산은 도로 네트워크를 관심지점 기준으로 표현하기 위한 PC(POI of Criterion)의 도메인과 PM, PCR, PTR과 같은 도로를 연결할 수 있는 Connection 연산을 통해 보행자의 이동 경로를 연결하고, 관심지점 기반의 경로 탐색을 위한 보행자 도로 네트워크의 MRT(Multi-Rule include Time) 연산과정으로 수행된다.

[정의 4] PC Domain

$Dom(PC) = \{(PC_ID, S_POI, ENTRY_OFFSET, ADJ_LIST) \mid PC_ID \in INT, S_POI \in POI_INFO, ENTRY_OFFSET \in INT[], ADJ_LIST \in \{PM \vee PCR \vee PTR\}\}$

PC의 도메인은 고유 아이디인 PC_ID, 시작 위치를 나타내는 S_POI, PM, PCR, PTR을 선택하여 인접 리스트에 접근할 수 있는 ENTRY_ROAD, 보행자 도로가 인접 리스트 형태로 구성된 ADJ_LIST로 이뤄진다.

[정의 5] 임의의 pc 가 PC의 집합에 속하고 임의의 p 가 관심지점의 집합에 속할 때, 링크 l 을 만들어내는 연산은 Connection이며 링크 $l = Connection(pc, p, road_type)$ 이다.

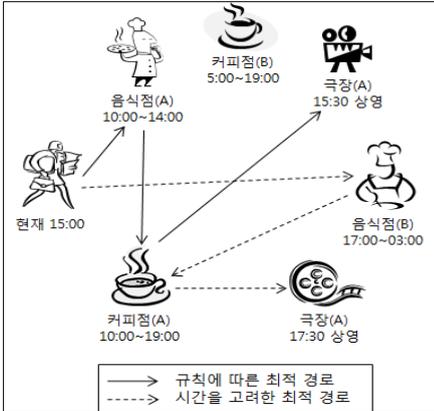
Connection 연산을 통해 보행자 길안내에서 목적지까지 경로의 결과를 알려주는 PR(Pedestrian Route)를 정의할 수 있다.

[정의 6] 임의의 pc_1, \dots, pc_n 이 PC의 집합에 속하고 임의의 p_1, \dots, p_n 이 관심지점의 집합에 존재할 때, PR은 $\sum_{i=1}^n Connection(pc_i, p_i)$ 의 결과이다.

PR은 목적지 관심지점을 통해 pc 의 인접 리스트에 저장된 도로를 선택하여 결과 값을 반환한다. 이 같은 방식으로 목적지까지 순차적인 결과 값들을 하나의 집합으로 나타내면 최종 목적지까지의 경로가 나타난다.

이 같은 도로 연산을 바탕으로 관심지점 기반의 경로 질의를 처리하기 위한 MRT 연산은 서로 다른 관심지점

들을 연결하여 경로를 안내하도록 돕는 보행자 길안내의 연산 타입이다. 또한, 상점의 개점 시간과 폐점 시간과 같은 시간 속성을 처리 하여 시간 속성을 고려한 관심지점을 선택하여 보행자의 경로를 안내한다.



[그림 1] 보행자 규칙에 따른 가능 경로 안내

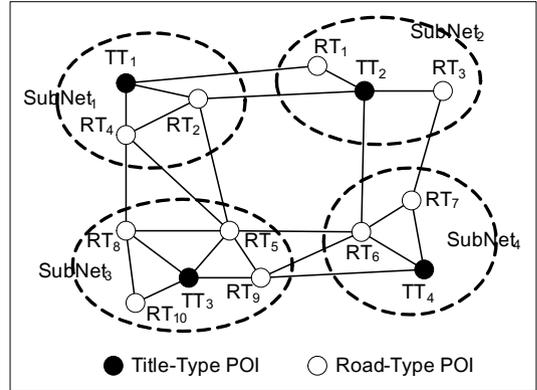
그림 1은 보행자 이동 경로 규칙에 따라 가능한 경로 안내를 나타낸다. 보행자는 음식점, 커피점, 극장 순으로 이동경로의 규칙을 정하고 목적지까지 경로 안내를 원한다. 보행자의 규칙에 따라 두 개의 가능한 경로가 존재한다. 두 가지 이동 가능한 경로의 차이점은 시간 속성을 고려하지 않은 경우와 시간 속성을 고려한 경우이다. 보행자는 오후 3시에 극장까지의 정해진 규칙에 따라 시간을 안내 받기를 원한다. 음식점, 커피점은 개점과 폐점 시간이 존재하고 최종 목적지인 극장의 상영 시간이 존재하기 때문에 짧은 거리 위주로 Solid 선을 따라 갈 경우 음식점의 이용과 영화 상영시간에 늦게 된다. 하지만, 현재 시간과 상점의 개점과 폐점 시간을 고려할 경우 영화 상영시간에 맞춰 경로를 안내 받을 수 있다.

이 같은 예를 통해 보행자 경로 규칙의 집합 C_i 가 적당한 관심지점의 카테고리를 포함하고 C_i 가 집합을 이룰 때, C_i 는 보행자 경로 규칙의 집합으로 정의할 수 있다. 보행자 경로 규칙의 집합을 기반으로 MRT 연산은 임의의 질의 Q 가 명시된 보행자 경로 규칙의 집합을 C_i (C_1, C_2, \dots, C_n) 이고 현재 시간을 T_c 를 포함하여 처리할 수 있는 연산으로 정의한다.

3.2 보행자 도로 네트워크 저장 관리

관심지점은 횡단보도, 육교와 같은 도로 관심지점 (RoadType-POI : RT-POI)와 음식점, 은행과 같은 명칭 관

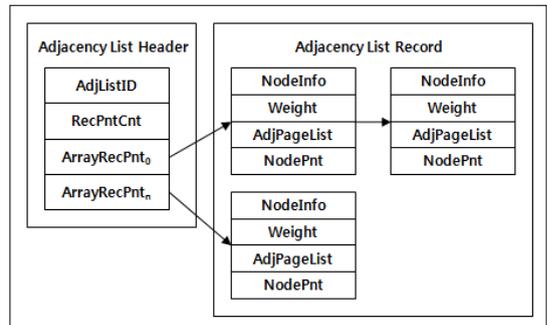
심지점(TitleType-POI : TT-POI)타입으로 나눈다. TT-POI는 H-POI를 통해 관리되며 보행자 도로 네트워크 그룹화는 TT-POI를 중심으로 주변의 도로 RT-POI를 포함하면서 그룹화 한다.



[그림 2] 보행자 도로 네트워크 그룹화

그림 2는 관심지점 기반 도로 네트워크 그룹화를 위해 TT-POI인 TT1, TT2, TT3, TT4를 기준으로 서브 네트워크를 분류한다. TT1은 인접한 RT-POI인 RT4와 RT2를 포함하여 SubNet1으로 그룹화 된다. 나머지 TT2, TT3, TT4도 TT1과 마찬가지로 SubNet2, SubNet3, SubNet4로 분류된다.

관심지점을 기반으로 도로 네트워크 그룹화 작업이 선행된 후 각 서브 네트워크는 인접 리스트로 구축된다. TT-POI를 기준으로 구성된 인접 리스트는 인접한 RT-POI를 리스트 형태로 갖고 있기 때문에 인접한 서브 네트워크가 서로 다른 데이터 페이지에 있을 경우 관심지점을 다시 검색하여 데이터 페이지에 접근하므로 발생하게 되는 추가 검색 시간을 방지하기 위해 인접 리스트의 자료구조는 아래 그림 3과 같이 구성한다.

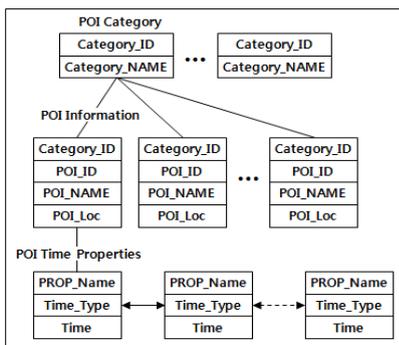


[그림 3] 인접 리스트의 구조

위 그림 3에서 인접 리스트 구조는 인접 리스트 헤더와 인접 리스트 레코드로 구성되어 있다. 인접 리스트 헤더는 인접 리스트를 식별할 수 있는 ID를 표현하는 AdjListID, 인접 리스트의 포인터 개수를 나타내는 RecPntCnt, 실제 인접 리스트의 레코드를 관리하는 배열 구조로 되어 있는 ArrayRecPnt로 구성되어 있다. 인접 리스트 레코드는 RT-POI 정보를 나타내는 NodeInfo, TT-POI와 RT-POI의 거리 가중치를 나타내는 Weight, 보행자 도로에 대한 정보를 갖고 있는 인접한 서브 네트워크가 서로 다른 데이터 페이지에 있을 때 레코드의 정보만으로 데이터 페이지를 찾는 AdjPageList, 다음 노드로 접근 가능한 NodePnt로 구성되어 있다.

관심지점 기반의 보행자 도로 네트워크의 인덱스는 H-POI(Hierarchical-POI)를 이용한다. H-POI는 관심지점 기반의 경로 질의가 입력된 경우 만족하는 관심지점에 대한 데이터 페이지의 빠른 탐색을 위한 인덱스 구조이다. H-POI는 POI_Category, POI_INFO, POI_TIME의 세 객체를 갖는다. POI_Category는 의미적, 공간적으로 분류된 관심지점을 관리하기 위해 다수의 POI_INFO 객체를 관리한다. POI_INFO는 각 관심지점 객체의 정보를 나타내며 관심지점의 시간정보를 리스트 구조로 관리한다. POI_PROPERTIES는 관심지점의 속성정보를 나타낸다.

그림 4는 H-POI의 객체 구조를 나타낸다. POI_Category는 보행자가 MRT 질의가 입력될 때 보행자 경로 규칙을 정하기 위해 사용 된다. POI_Category는 카테고리 고유의 ID를 나타내는 Category_ID와 각 관심지점 객체들을 의미적으로 통합될 수 있는 이름을 부여하기 위한 Category_Name으로 구성된다.

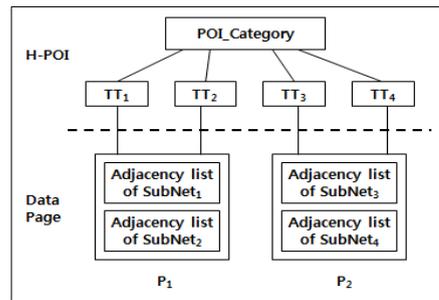


[그림 4] H-POI의 객체 구조

POI_INFO는 자신의 상위 객체인 POI_Category의 Category_ID와 모든 관심지점 고유의 ID인 POI_ID, 관심지점의 명칭을 나타내는 POI_Name와 실제 지도상에 위도와 경도를 표시하여 관심지점의 위치를 나타내는

POI_Loc으로 구성되어 있다.

POI_PROPERTIES는 상위 객체인 POI_INFO에 표시되어 있는 관심지점 고유의 ID, 변하지 않는 시간 ST(Static_Time)과 주기적으로 갱신되는 시간 DT(Dynamic_Time)인지를 판단하는 Time_Type과 실제 시간을 나타내는 Time으로 구성되어 있다.



[그림 5] 보행자 도로 네트워크 파일 구조

그림 5는 관심지점 기반의 보행자 도로 네트워크 파일 구조를 나타낸다. 보행자 도로 네트워크 파일 구조는 데이터 페이지에 빠르게 접근할 수 있는 H-POI와 실제 인접 리스트가 저장된 데이터 페이지로 구성된다.

3.3 보행자 도로 네트워크 알고리즘

관심지점 기반의 보행자 도로 네트워크의 관리를 위해 지원되는 연산은 관심지점의 벌크 삽입, 삽입, 삭제, 검색 연산으로 분류한다. 관심지점 검색 연산은 H-POI의 카테고리 고리를 검색한 후 관심지점의 정보와 일치하는 데이터 페이지에 접근하여 수행되며, 아래에서는 벌크 삽입, 삽입, 삭제에 대해 기술한다.

관심지점의 벌크 삽입 연산은 관심지점의 집합을 입력으로 그룹화, 인접 리스트 생성 및 데이터 페이지 삽입을 한 번에 처리하는 장점이 있으며, 벌크 삽입 연산은 아래 알고리즘 1에 따라 수행된다.

알고리즘 1에서는 TT-POI와 RT-POI를 분류하는 작업(줄1-6)과 각 TT-POI의 인접한 RT-POI들을 찾아 인접 리스트를 만들고 데이터 페이지에 저장하는 과정(줄7-14)을 나타낸다.

삽입 연산은 도로의 구획 변경에 따른 보행자 도로나 관심지점이 추가될 경우 이용하며, 처리 절차는 아래 알고리즘 2와 같다. 알고리즘 2에서 H-POI에서 POI의 데이터 위치를 검색하여 데이터 페이지를 반환한다(줄1). 데이터 페이지가 존재하고 사용 가능한 공간이 있을 경우 POI의 인접 리스트를 삽입한 후 H-POI의 인덱스 엔트리에 삽입하고 데이터 페이지의 주소 정보를 가진다(줄

2-4). 데이터 페이지가 존재하지만 데이터 페이지의 가용 공간이 없을 경우와 데이터 페이지가 존재하지 않을 경우 새로운 데이터 페이지를 할당한 후 데이터를 삽입하고(줄6-8), 삽입 POI가 인접한 POI와 데이터 페이지가 근접하도록 데이터 페이지를 재구성한다.

<알고리즘 1- 관심지점 벌크 삽입 알고리즘>

```

Algorithm 1 : BulkInsert()
Input   set of PV : POI의 집합
Output  ERR_POI : TT_POI, RT_POI가 아닌 경우
        ERR_FLUSH : 디스크 쓰기 실패
Begin
01 : for (i = 0; i <= |PV|; i++) do
02 :     if( PV(i) == TT_POI) then
03 :         add to list of TT_POI ;
04 :     else if( PV(i) == RT_POI) then
05 :         add to list of RT_POI ;
06 :     else return ERR_POI;
07 : for (i = 0; i <= |List of TT_POI|; i++) do
08 :     nnRT_POIList=findNNPOI(TT_POI(i),RT_POI);
09 :     if (nnRT_POIList != NULL) then
10 :         adj_List=createAdjList(TT_POI(i),nnRT_POIList);
11 :         if(remainSize(P) >= SIZE(adj_List)) then
12 :             add to P(adj_List);
13 :         else createNewPage(), add to P(adj_List);
14 : return if flush(P) is fail return ERR_FLUSH ;
End
    
```

<알고리즘 2- 관심지점 삽입 알고리즘>

```

Algorithm 2 : Insert(POI, List(POI))
Input   POI : TT-POI의 정보
        List(POI) : POI의 인접 리스트
Output  ERR_FLUSH : 디스크 쓰기 실패
Begin
01: P = findIndex(POI);
02 : if( P != null && P is available) then
03 :     insert List(POI) into data page P
04 :     insert H-POI index entry(POI, address(P))
05 : else if((P!=null&&P is not available) or P==null) then
06 :     P = createNewPage();
07 :     insert List(POI) into data page P
08 :     insert H-POI index entry(POI, address(P))
09 :     ReconstructionPage(P);
10 : return if flush(P) is fail return ERR_FLUSH ;
End
    
```

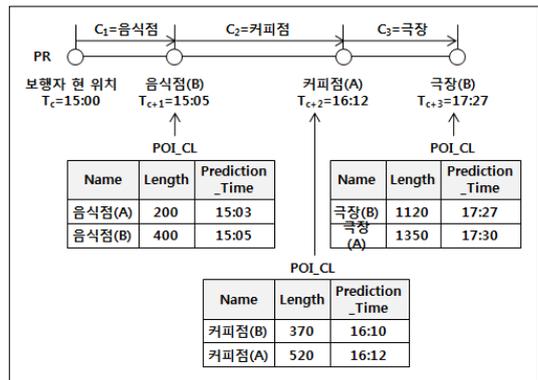
삭제 연산은 관심지점이 제거된 경우 사용되며, 수행 절차는 아래 [알고리즘 3]과 같다. 알고리즘 3에서 관심지점의 데이터 페이지가 존재할 경우 POI 정보를 삭제한다(줄2-8). POI와 인접한 RT-POI의 집합을 다른 TT-POI의 데이터 페이지에 저장한다(줄3-4). RT-POI를 다른 TT-POI에 저장하는 기준은 각 RT-POI와 최소 거리에 있는 POI를 기준으로 데이터 페이지에 저장한다.

<알고리즘 3- 관심지점 삭제 알고리즘>

```

Algorithm 3 : Delete(POI)
Input   POI : TT-POI의 정보
Output  ERR_POI : POI 정보 없음
        ERR_FLUSH : 디스크 쓰기 실패
Begin
01 : P = findIndex(POI);
02 : if(P != null) then
03 :     for(i = 0; i < P.List(POI).RecPntCnt; i++) do
04 :         insert List record
            into minimum distance Adjacency Page
05 :     delete P.List(POI);
06 :     delete index entry(POI);
07 :     ReconstructionPage(P);
08 : else
09 :     return ERR_POI;
10 : return if flush(P) is fail return ERR_FLUSH ;
End
    
```

3.4 경로 질의 처리



[그림 6] 경로 질의 처리 과정

경로 질의는 보행자의 경로 규칙에 대한 정보와 현재 시간 정보의 입력될 경우 MRT 연산을 수행한다. MRT

연산은 보행자의 이동 경로 규칙에 따라 현재의 위치에서 인접 지역 탐색 연산을 통해 카테고리에 맞는 관심지점을 찾아내고 찾아낸 관심지점의 위치에서 사용자가 다음으로 정한 카테고리로 관심지점을 검색하여 목적지까지의 보행자 도로를 나타낸다.

그림 6은 그림 1의 시나리오를 기반으로 구성된 경로 질의 처리 과정의 예이다. 그림 6에서 보행자의 평균 이동 속도 V 는 $1.2m/s$ 이며 한 지점의 관심지점을 기준으로 보행자의 체류시간은 기본 1시간으로 가정한다. 또한, 보행자 이동 경로 규칙의 집합 C_i 는 그림 3-3과 같이 음식점, 커피전문점, 극장 순이며 현재 시간은 오후 3시로 가정한다.

보행자 이동 경로 규칙으로 선행되는 작업은 현재 위치를 중심으로 C_1 의 규칙인 음식점 카테고리를 검색하여 POI_CL(POI Candidate List)을 뽑는다. POI_CL를 뽑는 첫 번째 기준은 보행자가 정한 규칙의 카테고리 안에 포함되는지 여부를 판단하게 된다. 두 번째 기준은 두 관심지점의 거리를 계산하여 짧은 거리를 우선으로 순위를 부여받게 된다. POI_CL은 관심지점의 이름을 나타내는 Name, 관심지점의 거리를 계산하는 Length, 도착지 관심지점까지의 예상 도착 시간을 표현한다. 예상 도착 시간은 미리 정의된 보행자의 평균 이동 속도와 거리를 이용하여 예상 이동 시간과 미리 기본 값으로 정의된 관심지점의 체류 예상 시간을 합한 시간으로 계산한다.

보행자의 현재 위치에서 POI_CL은 A 음식점, B 음식점 순이다. A 음식점까지의 예상도착 시간은 오후 3시 3분이며 B 음식점까지는 오후 3시 5분이다. 하지만, A 음식점의 폐점시간이 오후 2시인 것을 고려하여 A 음식점은 최종 경로에서 탈락하고 B 음식점을 선택한다. B 음식점은 경로 탐색을 위한 기준이 되고 커피 전문점 카테고리를 검색하여 인접한 커피 전문점의 POI_CL을 선택한다. 마찬가지로 예상시간을 계산하게 되는데 현재의 관심지점에서의 체류 예상 시간을 합하여 계산하여 거리상으로 가장 짧은 위치와 현재 영업 중인 A 커피 전문점을 선택한다. 이 같이 순차적으로 관심지점을 선택하여 최종적으로 사용자가 정한 카테고리에 의해 B 극장이 최종 목적지로 선택되게 되고 경로를 안내하게 된다.

4. 성능 평가

성능 평가에 사용된 시스템 환경은 RIM(Research In Motion)사의 BlackBerry Bold 9000을 사용했으며 CPU는 624MHz이고 메모리 용량은 1GB이다. BlackBerry Bold

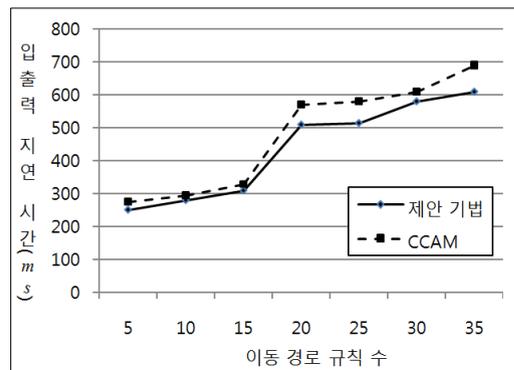
9000에서 사용된 모바일 전용 운영체제는 RIM 에서 자체 개발한 BlackBerry 운영체제이며 어플리케이션 개발을 위한 언어는 Java이다. 보행자 길안내 개발을 위해 JDE 4.6.0 버전을 이용하고, 지도 데이터는 루센 맵을 이용해 구현하였다[17].

성능 평가의 경로 질의는 사용자 선호도에 따라 관심지점 카테고리 규칙을 정하여 규칙의 순서에 따라 처리 가능한 MRPSR[15] 질의를 이용하였고, 실험은 시뮬레이션 데이터를 이용하여 질의를 처리하고 그 결과를 BlackBerry에서 검증하는 방식으로 진행하였다.

4.1 디스크 입출력 비용

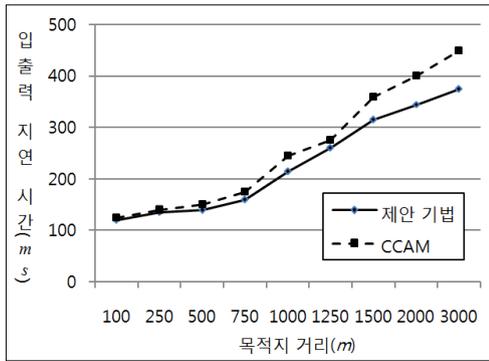
디스크의 입출력 시간은 관심지점 데이터 페이지를 찾기 위해 디스크를 탐색하는 시간과 입출력 횟수와 디스크의 입출력 시간을 곱한 값을 합한 시간을 말한다. MRPSR 질의 이동 경로 규칙의 개수에 대한 디스크 입출력 지연 시간은 B+ 색인을 이용한 CCAM을 대상으로 한다. 실험을 위해 시작과 목적지는 같은 위치를 선정하고 MRPSR 질의의 이동 경로 규칙 개수를 5개씩 늘려가는 실험으로 디스크 입출력 시간을 비교한다.

그림 7에서 CCAM는 관심지점 위주의 색인이 구축되어 있지 않아 사용자가 입력한 관심지점 카테고리 내에 있는 관심지점을 검색하기가 힘들다. 하지만, 제안기법인 관심지점 기반의 도로 네트워크 저장기법은 H-POI를 이용한 인덱스를 통해 디스크의 데이터 페이지에 직접적으로 접근이 가능하기 때문에 CCAM에 비해 지연이 감소함을 알 수 있다.



[그림 7] 이동경로 규칙 수에 따른 입출력 지연시간

다음은 목적지 거리에 따라 경로 질의 처리 시 입출력 지연 시간을 비교한다. 실험을 위해 목적지의 위치는 동일한 위치로 정하고 100m, 250m, 500m, 750m, 1km, 1.25km, 1.5km, 2km, 3km 순으로 실험을 진행하였다.

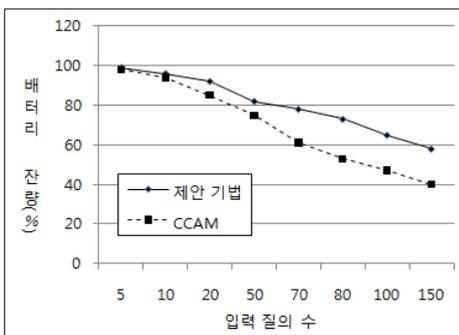


[그림 8] 목적지 거리에 따라 경로 질의 처리 시 디스크 입출력 지연시간

그림 8에서 CCAM에 비해 제안기법이 목적지까지 거리에 따라 디스크 입출력 지연 시간이 단축되었음을 알 수 있다. CCAM이 제안기법에 비해 느린 이유는 경로 탐색 질의 처리 시 데이터 페이지에 접근하기 위해 인덱스의 재탐색 시간이 추가적으로 발생한다. 반면, 제안기법은 H-POI로 구축된 인덱스에 한 번만 접근하면 인접 리스트 내에 데이터 페이지의 위치 정보를 갖기 때문에 다른 데이터 페이지에 바로 접근이 가능하다.

4.2 전력 소모량 평가

MRPSR 질의 처리 시 입출력 횟수에 따른 배터리 소모량을 평가하기 위해 경로 탐색 질의의 횟수를 늘려가면서 배터리 소모량을 확인하였다.



[그림 9] 질의 입력 횟수에 따른 배터리 소모량

그림 9에서 경로 탐색 질의 입력 횟수를 5번에서 150번까지 입력할 때 배터리 잔량을 나타낸다. CCAM은 제안기법에 비해 디스크의 입출력 횟수가 많기 때문에 배터리 소모량이 크다. 제안기법은 기존 기법에 비해 약 20% 정도의 배터리를 절약한다.

5. 결론

본 논문에서는 스마트폰에서 효과적인 보행자 길안내를 위해 보행자 도로 네트워크 모델 및 저장 방법, 그리고 관심지점 기반의 경로 탐색 기법을 제안하였다. 제안 기법에서는 보행자 도로 네트워크 모델을 통해 기존의 차량을 위한 도로 네트워크 모델 기반의 길안내에서는 불가능한 실제 보행자가 이동하는 경로 정보를 이용하여 보다 정확한 보행자 길안내가 가능하게 하였고, 경로 탐색에 있어 관심지점 정보를 기반으로 하는 인덱스를 구축하여 경로 질의 처리 시 기존의 기법보다 디스크 입출력 비용을 약 10% 감소시켰으며 이로 인한 전력 소모량도 20% 절약할 수 있음을 보였다. 향후 연구로 보행자 길안내에서 경로 질의 처리 속도 향상을 위한 알고리즘에 대한 연구와 주기적으로 시간에 따라 위치가 변화하는 이동 객체들을 고려한 경로 탐색 질의에 대한 연구가 필요하다.

참고문헌

- [1] 조숙경, 김성희, 김종훈, "u-GIS 환경에서 플랫폼 독립을 지원하는 플렉스 기반 모바일 지리 정보 서비스 시스템의 설계 및 구현", 한국공간정보시스템학회 논문지, 제 1권 제1호, pp. 87-95, 2009.
- [2] 유지은, "스마트폰의 Key Enabler: 소프트웨어", 정보통신산업진흥원, 2009.
- [3] Christian S. Jensen, Torben Bach Pedersen, Laurynas Speicys, and Igor Timko, "Data Modeling for Mobile Services in the Real World", LNCS 2750, pp. 1-9, 2003.
- [4] 김돈한, "보행자의 감성을 고려한 경로탐색 지원 시스템 제안", 한국디자인학회 제64호, pp81-90, 2006.
- [5] Ayes Ozbil, John Peponis, "Modeling Street Connectivity And Pedestrian Movement According to Standard GIS Street Network Representations", of Proc 6th ISSS, 2007.
- [6] Randall, T.A., Baetz, B.W., "Evaluating Pedestrian Connectivity For Suburban Sustainability", Journal of Urban Planning and Development, vol.127, no.1, pp.1-15. 2001.
- [7] Sang Ki Kim, Sung Ha Baek, Dong Wook Lee, Warn Il Chung, Gyoung Bae Kim, Hae Young Bae, "Data Source Management using weight table in u-GIS DSMS", 한국공간정보시스템학회, 제11권 2호, pp 27-33, 2009.

- [8] Shashi Shekhar, Duen-Ren Liu, "CCAM: A Connectivity-Clustered Access Method for Aggregate Queries on Transportation Networks : A Summary of Results", In Proc. of the ICDE, 1995.
- [9] Mehdi Sharifzadeh, Mohammad Kolahdouzan, Cyrus Shahabi, "The optimal sequenced route query", VLDB Journal, pp. 765-787, 2008.
- [10] Feifei Li, Dihan Cheng, Marios Hadjieleftheriou, George Kollios, Shang-Hua Teng, "On Trip Planning Queries in Spatial Databases", of Proc. 9th SSTD, pp. 273-290, 2005.
- [11] Haiquan Chen, Wei-Shinn Ku, Min-Te Sun, Roger Zimmermann, "The Multi-Rule Partial Sequenced Route Query", In Proc. of the ACM GIS, 2008.
- [12] Ralf hartmut Guting Victor Teixeira de Almeida, Zhiming Ding, "Modeling and Querying moving objects in networks", VLDB Journal, pp. 165-190, 2006.
- [13] Sungwon Jun, Sakti Pramanik, "HiTi Graph Model of Topographical Road Maps in Navigation Systems", In Proc. of Data Engineering, pp. 76-84, 1996.
- [14] Dimitris Papadias, Jun Zhang, Nikos Mamoulis, Yufei Tao, "Query Processing in Spatial Network Databases", of Proc. the 29th VLDB Conference, pp. 802-813, 2003.
- [15] Sung-Ho Woo, Sung-Bong Yang, "A Linear-Order Based Access Method for Efficient Network Computations", LNCS 2480, pp. 289-302, 2002.
- [16] Cruz, I.F., Mendelzon, A.O., Wood, P.T, "A Graph-oriented object database model", In Proc. ACM Conference on PODS, pp.417-424, 1990.
- [17] Research In Motion, "BlackBerry JDE Plug-in for Eclipse:Getting started guid", <http://na.blackberry.com/eng/developers/javaappdev/#>, 2009.

정 원 일(Weonil Chung)

[정회원]



- 1998년 2월 : 인하대학교 전자계산공학과 (공학사)
- 2004년 8월 : 인하대학교 컴퓨터정보공학과 (공학박사)
- 2004년 7월 ~ 2006년 7월 : 한국전자통신연구원 선임연구원
- 2007년 3월 ~ 현재 : 호서대학교 정보보호학과 교수

<관심분야>

데이터스트림, 이동객체, 시스템보안

김 상 기(Sang-Ki Kim)

[정회원]



- 2007년 2월 : 서원대학교 컴퓨터교육과 (학사)
- 2008년 3월 ~ 현재 : 인하대학교 컴퓨터정보공학과 석사과정

<관심분야>

LBS, 데이터스트림, 공간데이터베이스