

# 도로 네트워크에서 A\* 알고리즘을 이용한 k-최근접 이웃 객체에 대한 효과적인 경로 탐색 방법

(Efficient Path Finding Based on the A\* algorithm for Processing k-Nearest Neighbor Queries in Road Network Databases)

신 성 현 <sup>†</sup> 이 상 철 <sup>\*\*</sup> 김 상 욱 <sup>\*\*\*</sup> 이 정 훈 <sup>\*\*\*\*</sup> 임 을 규 <sup>\*\*\*\*\*</sup>  
 (Sung-Hyun Shin) (Sang-Chul Lee) (Sang-Wook Kim) (Junghoon Lee) (Eul Gyu Im)

**요 약** 본 논문에서는 기존 k-최근접 객체 검색의 효율성을 개선하고 도로 네트워크에의 응용을 용이하게 하기 위하여 질의 점으로부터 k개의 정적 객체까지의 경로를 효과적으로 탐색할 수 있는 방법을 제안한다. 제안한 방법은 우선, k-최근접 이웃 질의 방법을 이용하여 후보 정적 객체들을 선정한 후 이들 후보 객체들의 위치 정보를 이용하여 최단 경로를 탐색한다. 일대다 경로탐색을 위하여 A\* 알고리즘을 개선하여 반복된 일대일 경로탐색에 따르는 중복된 노드 스캔을 제거한다. 또, 계산된 결과를 이용하여 질의 점으로부터 네트워크 거리상으로 가까운 k개의 정적 객체들의 위치를 재정렬하여 반환한다. 성능평가 실험 결과, 제안한 방법은 기존 방법들인 INE, post-Dijkstra, 그리고 naïve method에 비해 정확성이 100%로 매우 높게 나타났으며, 노드 탐색 시간은 1.3~3.0배로 향상된 성능을 보였다.

**키워드** : 도로 네트워크, k-최근접 이웃 질의, A\* 알고리즘

**Abstract** This paper proposes an efficient path finding scheme capable of searching the paths to k static objects from a given query point, aiming at both improving the legacy k-nearest neighbor search and making it easily applicable to the road network environment. To the end of improving the speed of finding one-to-many paths, the modified A\* obviates the duplicated part of node scans involved in the multiple executions of a one-to-one path finding algorithm. Additionally, the cost to the each object found in this step makes it possible to finalize the k objects according to the network distance from the candidate set as well as to order them by the path cost. Experiment results show that the proposed scheme has the accuracy of around 100% and improves the search speed by 1.3~3.0 times of k-nearest neighbor searches, compared with INE, post-Dijkstra, and naïve method.

**Key words** : road network, k-nearest neighbor query, A\* algorithm, branch-and-bound

· 본 논문은 2008년도 정부(교육과학기술부)의 재원으로 한국과학재단의 지원 (R01-2008-000-20872-0) 및 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업(IITA-2009-C1090-0902-0040)의 지원을 받아 수행된 연구임

<sup>†</sup> 정 회 원 : 한양대학교 공과대학 컴퓨터공학부 Post-Doc.

brainshin@gmail.com

<sup>\*\*</sup> 학생회원 : 한양대학교 공과대학 컴퓨터공학부

korly@agape.hanyang.ac.kr

<sup>\*\*\*</sup> 중신회원 : 한양대학교 공과대학 컴퓨터공학부 교수

wook@hanyang.ac.kr

<sup>\*\*\*\*</sup> 비 회 원 : 제주대학교 전산통계학과 교수

jhlee@jeju.ac.kr

<sup>\*\*\*\*\*</sup> 비 회 원 : 한양대학교 공과대학 컴퓨터공학부 교수

imeg@hanyang.ac.kr

논문집수 : 2009년 7월 19일

심사완료 : 2009년 9월 10일

Copyright©2009 한국정보과학회 : 개인 목적이거나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 데이터베이스 제36권 제5호(2009.10)

### 1. 서론

최근 이동 객체의 위치 기반 서비스(Location-Based Services: LBS)를 지원하는 응용 분야는 개인이나 차량의 위치를 파악하여 차량 항법, 교통 정보와 같은 부가 서비스를 제공한다[1]. LBS를 제공하는 실제 응용은 차량 등과 같은 이동 객체와 주유소 등과 같은 정적 객체의 위치 정보를 효율적으로 검색하여야 하는데, 이를 위해  $k$ -최근접 이웃( $k$ -Nearest Neighbors:  $k$ -NN) 질의 처리가 필수적으로 요구된다.  $k$ -NN 질의란 질의가 요청한 위치로부터 공간 상에서 가장 가까운  $k$ 개의 정적 객체를 검색하고 이를 반환하는 질의이다[2,3].

$k$ -NN 질의 처리를 위한 연구로는 우선, Papadias 등은 유클리드 거리가 항상 네트워크 거리보다 작거나 같을 수밖에 없다는 특성에 착안하여 IER(Incremental Euclidean Restriction) 방법을 제안하였다[3]. 이와 더불어 모든 도로 세그먼트를 점진적으로 확장하면서 정적 객체를 검색하는 INE(Incremental network expansion) 방법을 제안하였다. 이들 방법은 저장 비용이 작다는 장점에 비해, 질의 처리시 여러 번의 시행착오가 수반되거나, 색인 관리 오버헤드가 발생하는 문제점을 갖고 있다.

다음으로, Kolahdouzan 등은 도로 네트워크를 작은 셀 단위로 나누고, 나누어진 셀의 정보를 이용하여 정적 객체들의 위치를 미리 계산(pre-compute)하는  $VN^3$  방법을 제안하였다[4]. 그리고, Lee 등은 네트워크 거리를 기반으로 FastMap[5]을 이용하여 정적 객체들을 인덱싱하고, 이를 이용하여 효과적으로 질의 처리하는 근사 인덱싱 방법을 제안하였다[2]. 이들 방법은 IER과 INE 방법보다 우수한 검색 성능을 보이지만, 질의 점과 가까운  $k$ 개의 정적 객체들을 찾을 뿐, 네트워크 경로를 탐색해주지 않는 문제점을 갖고 있다.

본 논문에서는 질의 점으로부터  $k$ -NN 정적 객체를 검색한 후, 이들에 대한 효율적인 최단 네트워크 거리 및 경로를 계산하는 방법을 제안하고 성능을 평가한다. 이러한 방법은 먼저, 네트워크 거리를 기준으로 가까운  $k$ 개의 정적 객체를 찾는 단계부터 시작하며,  $VN^3$ 와 근사 인덱싱 방법과 같은  $k$ -NN 질의 처리 방법을 이용하여 정적 객체를 일차적으로 추출한다. 이전 과정에서 찾은  $k$ 개의 객체에 대한 경로를 계산하는 데, 기존  $A^*$  알고리즘을 개선하여 질의 점으로부터  $k$ 개의 정적 객체에 대한 각 경로를 통합적으로 탐색하는 방법을 제안한다. 마지막으로, 실제 구한 경로와 거리 정보를 이용하여 검색된  $k$ 개의 정적 객체를 질의 점으로부터 가까운 순서로 정렬하여 반환한다.

본 논문의 구성은 다음과 같다. 제2장에서는 본 논문

에서 다루는  $k$ -NN 질의 처리와 최단 경로 탐색을 위한 동작 프레임워크를 설명한다. 제3장에서는 제안하는 최단 경로 탐색 방법을 제시하고 관련 알고리즘을 설명한다. 제4장에서는 실험을 통해 제안한 방법의 우수성을 보이고, 마지막으로 제5장에서 결론을 맺는다.

### 2. $k$ -NN queries에 대한 프레임 워크

본 논문에서는  $VN^3$  방법과 근사 인덱싱 방법과 같은  $k$ -NN 질의 처리 방법을 이용하여 추출된 정적 객체들의 위치를 파악하고, 정확한  $k$ -NN 객체를 선별하기 위해 최적 경로를 효과적으로 탐색하는 방법을 제안한다.

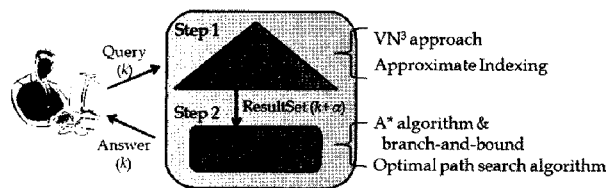


그림 1  $k$ -NN 질의 처리를 위한 프레임 워크

그림 1은  $k$ -NN 질의를 수행하고 결과를 반환하는 동작 프레임워크를 나타낸다. 먼저, 인덱스 부분은  $k$ -NN 질의 처리 방법을 이용하여  $k$ 개의 정적 객체들의 위치를 검색한다. 여기서,  $k$ -NN 질의 처리 방법은 검색 과정에서 객체간 거리 오차가 발생한다. 이러한 오차를 감소시키기 위해서는  $k$ 에 대한  $pseudo-k$ 의 비율( $pseudo-k/k$ )에 대해  $pseudo-k=k(1+a)$ 일 때  $a$ 는 0보다 큰 수행 변수로 검색하면 오차를 크게 감소시킬 수 있다[2].

다음으로, 후처리 부분은 인덱스 부분을 통해 검색한 정적 객체들의 최단 경로 및 거리를 계산하는 과정에서  $A^*$  알고리즘[6]을 개선하여 탐색한다. 여기서, 노드를 스캔하는 과정에서 불필요한 도로 세그먼트의 탐색을 감소시켜야 한다. 또한, 이 부분은 인덱스 부분에서 추출된  $k+a$ 개의 정적 객체들의 실제 네트워크 거리를 계산하여  $k$ 개의 정적 객체들로 선별하여 결과로 반환한다.

### 3. 최단 경로 검색 기술

#### 3.1 $A^*$ 알고리즘

기존의 연구들은 도로 네트워크에서 최단 경로를 탐색하는 방법들을 제안하였다. 그 중 대표적인 알고리즘은 Dijkstra 알고리즘[7]과  $A^*$  알고리즘[6]이다. 그림 2는 이들 방법에 대해 1-NN 질의 처리의 예를 보이고 있다. 그림 2(a)는 Dijkstra 방법으로 모든 도로 세그먼트를 탐색하므로 불필요한 노드 스캔 영역이 넓어지며, 이로 인해 성능을 저하시킨다. 반면에, 그림 2(b)는  $A^*$  알고리즘 방법으로 불필요한 노드 스캔을 감소시키므로 Dijkstra 방법에 비해 효과적인 경로 탐색을 수행할 수 있다.

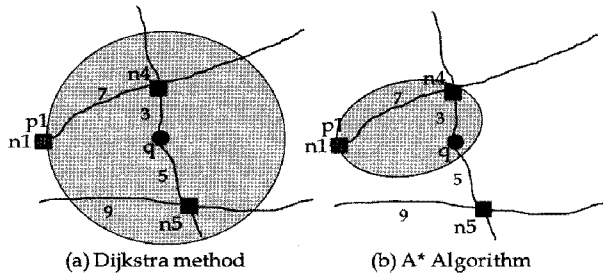


그림 2 1-NN 질의 처리를 위한 검색 영역의 예

이러한 효과적인 A\* 알고리즘은 도로 네트워크 상에서 비용이 낮은 노드를 탐색하기 위해,  $f$  (fitness),  $g$  (goal),  $h$  (heuristic) 속성들을 이용한 비용 함수는 다음 공식(1)과 같이 정의한다[6].

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

여기서,  $g(n)$ 은 도로 네트워크에서 시작 노드로부터 현재 노드  $n$ 까지의 경로 비용이고,  $h(n)$ 는 휴리스틱 값으로, 현재 노드  $n$ 에서 목표 지점까지 계산되는 휴리스틱 비용이다. 그러나, A\* 알고리즘은  $k$ -NN 질의를 처리하는 과정에서  $k > 1$ 일 때 중복된 노드 스캔이 발생하며, 노드의 수가 방대할 경우 탐색 영역이 넓어지는 문제가 있다.

### 3.2 제안 방법

본 논문에서는  $k$ 개의 객체들을  $k$ -NN 질의 처리 방법으로 빠르게 추출하고, 이들 정적 객체들의 경로 및 거리를 검색하는 방법을 제안한다. 질의 점으로부터  $k$ -NN 정적 객체를 검색하기 위해서는 최근접 정적 객체의 정형적인 정의가 필요하다. 이를 위해 본 논문에서는  $k$ 개의 정적 객체를 다음과 같이 정의한다.

**정의 1:** 도로 네트워크  $V$ 에서 질의 객체  $q \in V$ 와 정적 객체의 수  $k \geq 1$ 가 주어졌을 때,  $q$ 로부터 가장 근접한  $k$ 개의 객체들을  $q$ 의  $k$ -NN 정적 객체( $k$ -nearest neighbor static objects)라고 한다.

다음은  $k$ -NN 정적 객체들의 최단 경로를 나타내기 위해 두 정점  $u$ 와  $v$  간의 네트워크 거리 및 유클리드 거리를 다음과 같이 표현한다.

**정의 2:** 도로 네트워크  $V$ 에서 노드는  $u$ 와  $v$ 이고,  $u, v \in V$ 이다. 도로 세그먼트에서 노드  $u$ 는  $(u, v)$ 의 시작 노드이고,  $v$ 는  $(u, v)$ 의 끝 정점이라고 할 때, 이들의 네트워크 거리는  $dist_N(u, v)$ 이고, 유클리드 거리는  $dist_E(u, v)$ 라 정의한다.

**최단 경로 계산** 질의 점으로부터 임의의 정적 객체까지의 최단 경로를 탐색하기 위해 A\* 알고리즘을 개선하여 이용한다. 다음 정리 1은 A\* 알고리즘을 개선한 최단 경로식을 다음과 같이 정리한다.

**정리 1:** 도로 네트워크에서 현재 노드  $w$ 를 경유하는 노드  $u$ 와  $v$ 간의 최단 거리  $dist_{NE}(u, w, v)$ 라 정의하며,

이는 다음 공식 (2)와 같이 성립한다.

$$dist_{NE}(u, w, v) \leq dist_N(u, w) + dist_E(w, v) \quad (2)$$

**증명:** 시작 노드에서 현재 노드  $w$ 까지의 경로 비용은 네트워크 거리를 의미하므로 보조 정리 1에 따라  $dist_N(u, w)$ 가 성립된다. 다음으로, 현재 노드  $w$ 부터 목표 노드  $v$ 까지의 휴리스틱 비용은 보조 정리 2에 따라  $dist_E(w, v)$ 가 성립함을 알 수 있다. □

**보조 정리 1:** 시작 노드는  $u = w_0$ 이고 현재 노드는  $w = w_i$ 일 때,  $w_0$ 과  $w_i$ 의 연속으로 구성된 경로는  $w_0, w_1, \dots, w_{i-1}, w_i$ 이면, 네트워크 거리는 다음 공식 (3)을 만족한다.

$$\sum_{w_j \in V, j=1}^i dist(w_{j-1}, w_j) = dist_N(w_0, w_i) \quad (3)$$

**증명:** 현재 노드  $w_i$ 까지의 네트워크 거리 비용  $dist_N(w_0, w_i)$ 는 경유하는 각 노드 간의 거리가 누적된 거리 비용을 의미한다. 이에 따라,  $dist_N(w_0, w_i)$ 의 분할한 거리 비용은  $dist_N(w_0, w_1) + dist_N(w_1, w_2) + \dots + dist_N(w_{i-1}, w_i)$ 로 나타낼 수 있으며, 이는 다시  $\sum_{w_j \in V, j=1}^i dist_N(w_{j-1}, w_j)$ 가 되므로 공식 (3)이 성립한다. □

**보조 정리 2:** 현재 노드  $w$ 에 대해  $\forall w \in V$ 를 만족하고 목표 노드  $v$ 라 할 때, 휴리스틱 거리 함수  $h(w)$ 에 대하여 다음 조건이 성립한다.

$$h(w) \leq dist_E(w, v) \quad (4)$$

**증명:** 현재 노드  $w_i$ 에서 목표 노드  $v$ 까지의 휴리스틱 거리  $dist_E(w, v)$ 는 도로 네트워크 상에서 직선 거리인 유클리드 거리 비용을 의미하므로, A\* 알고리즘에서의 휴리스틱 거리  $h(w)$ 와의 관계가 성립한다. □

그림 3은 시작 노드  $u$ (또는  $u'$ )에서 목표 노드  $v$ (또는  $v'$ )까지의 최단 경로에 대한 예를 보이고 있다. 그림에서, 시작 노드  $u$ 와 중간 노드  $w$ 와의 네트워크 거리는  $dist_N(u, w)$ 이고, 목표 노드  $v$ 와의 유클리드 거리는  $dist_E(w, v)$ 이다. 중간 노드  $w$ 의 총 거리 비용은 정리 1에 따라  $dist_N(u, w) + dist_E(w, v)$ 로 계산할 수 있다. 여기서,  $dist_E(w, v)$ 는  $dist_N(w, v)$ 보다 거리 비용이 적으므로  $dist_E(w, v) \leq dist_N(w, v)$ 가 성립된다.

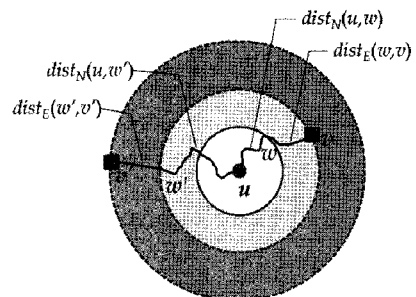


그림 3 전체 거리 비용 계산

**효과적인 최적 노드 선별** 본 논문에서는 A\* 알고리즘의 개선으로 최단 경로를 탐색할 경우에 경유하는 최적의 경로를 효과적으로 선별하기 위한 방법을 제안한다. 제안한 방법은 주어진 두 노드  $u$ 와  $v$ 를 연결하는 인접 노드  $w_i(1 \leq i \leq k)$  중에  $dist_{NE}(u, w_i, v)$ 의 비용이 가장 작은 노드를 탐색하는 것으로 불필요한 경로 탐색 비용을 줄이는 방법이다. 본 논문에서는 이러한 노드를 경유하는 최적 경로를 다음과 같이 정의한다.

**정의 3:** 도로 네트워크 상에서 주어진 두 정점  $u$ 와  $v$  사이에서 경유하는 여러 인접 노드들 중에서 정리 (1)에 따라 가중치의 합이 가장 적은 경로를 최적 경로(*optimal path*)라 정의한다.

최적 경로  $dist_{NE}^*(u, w_i, v)$ 의 비용은  $dist_{NE}(u, w_i, v)$ 의 비용보다 항상 작거나 같아야 한다. 최단 경로를 탐색하는 과정에서 주어진 노드와 연결된 여러 인접 노드들 중에 최적 노드를 탐색하기 위해 공식 (5)를 제시한다.

**정리 2:** 도로 네트워크에서 두 노드  $u$ 와  $v$ 가 주어질 때,  $u$ 에서  $v$ 로 경유하는 인접 노드들  $w_i(1 \leq i \leq k)$ 가 존재한다면 정의 3에 의한 최적의 경로는 다음 공식 (5)로 정리한다.

$$dist_{NE}^*(u, w_i, v) = \min_{1 \leq i \leq k} (dist_{NE}(u, w_i, v)) \quad (5)$$

**증명:** 특정 노드  $u$ 와 이와 여러 인접 노드들  $w_i$ 와  $w_j$ 가 존재한다면, 정의 1에 따라, 만일  $dist_N(u, w_i) + dist_E(w_i, v)$ 의 값이  $min$ 일 경우,  $dist_{NE}(u, w_i, v) < dist_{NE}(u, w_j, v)$ 가 된다. 그러므로  $dist_{NE}^*(u, w_i, v)$ 는 노드  $u$ 에서  $v$ 로의 경로들 중에서  $w_i$ 로 경유하는 최적 경로이다. □

**$k$ 개의 노드까지의 효과적인 경로 탐색** 그림 4는 특정 노드까지 효과적인 경로 탐색의 예를 보이고 있다. 질의 점  $q$ 로부터 정적 객체  $o1$ 까지의 경로를 탐색하기 위해, 먼저 질의 점  $q$ 로부터 노드  $n4, n5$ 를 탐색하고, 공식 (2)를 이용하여  $dist_{NE}(q, n4, n1)$ 와  $dist_{NE}(q, n5, n1)$ 를 계산한다. 이를 다시, 공식 (5)를 이용하여 최적의 노드를 탐색하면,  $dist_{NE}^*(q, n4, n1)$ 가 계산되어 이를 힙에 저장한다. 이러한 과정을 다시 반복하는 과정으로 모든 경로를 탐색한다. 이때, 기존에 힙에 저장된 정보를 사용하여 경로를 탐색한다. 이러한 효과적인 경로 탐색 과정은 최단 경로 탐색으로 선정된 특정 노드와 사전에 예측한  $k$ 개의 객체 사이에 최소 거리를 계산함으로써, 객체들과의 최단 경로 탐색과 객체의 근접한 거리를 계산할 수 있다.

그림 5는  $k$ -최근접 이웃 질의를 처리하기 위한 알고리즘을 나타낸다. 라인 1에서,  $k$ -NN 질의 처리 방법에 의해 추출된  $pseudo-k$ 개의 정적 객체들  $\{o1, o2, \dots, o_{pseudo-k}, \dots, o_k\}$ 을 객체 데이터 집합  $S$ 에 저장한다. 라인 2에서는 질의 점  $q$ 가 위치한 도로 세그먼트  $n_w, n_w$ 를 찾는다. 라인 3과 4에서는 세그먼트  $n_w, n_w$  상에 모든 정적 객체들을 검

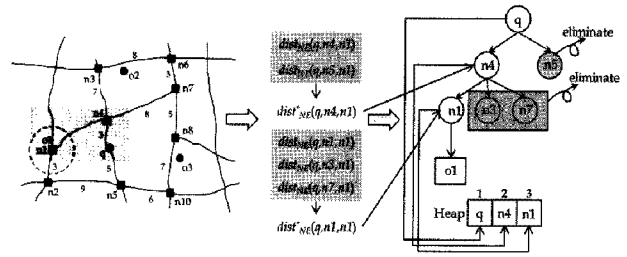


그림 4 힙을 이용한 효과적인 경로 탐색

```

Input: query point q, and pseudo-k nearest objects set
S={o1, o2, ..., Opseudo-k, ..., Ok}
Output: paths to each of k objects
1. S={o1, o2, ..., Opseudo-k, ..., Ok} by the k-NN search scheme
   underlying index structure;
2. find the segment  $n_w, n_w$  that covers q;
3. search all the objects on the segment  $n_w, n_w$ ;
4.  $H := \langle (n_i, dist_{NE}(q, n_i)), (n_w, dist_{NE}(q, n_w)) \rangle$ ;
5. for each object k do
6. de-heap the node n in Q with the minimum cost of
    $dist_{NE}(q, n_i, n_i)$ ;
7. for each non-visited adjacent node  $n_i$  of  $n_k$  do
8. if then  $dist_N(q, n_{i+1}) + dist_E(n_{i+1}, n_w) < dist_{NE}(q, n_i, n_w)$  then
9.    $dist_N(n_i) := dist_N(n_{i+1}) + dist_E(n_{i+1}, n_i)$ ;
10. if  $o_k := search$  all objects on the segment  $n_i, n_{i+1}$  then
11.   update  $\{o1, o2, \dots, o_k\}$  with sorted S;
12.   en-heap  $(n_i, dist_{NE}(q, n_i, n_i), o_k)$ ;
13. end if
14. end if
15. end for
16. end for
    
```

그림 5  $k$ -NN 질의 처리 알고리즘

색한 후, 질의와 가까운 순으로 힙  $H$ 에 저장한다. 라인 5~15에서는 인접한 다중 노드를 분기 한정 단계를 거쳐 최단 경로를 검색하는 작업을 수행한다. 먼저, 라인 6에서는 힙  $H$ 에서 다음 노드를 추출하여 도로 세그먼트를 확장한다. 그런 후, 라인 7~9에서는 인접한 다중 노드 중에 최단 거리가 짧은 경로를 찾기 위한 과정을 수행한다. 라인 10에서는 확장된 도로 세그먼트  $n_i, n_{i+1}$ 에 사전에 검색된 정적 객체가 존재하면 결과 집합에 추가한다. 마지막으로, 라인 11~12에서는 검색된  $k$ 번째 정적 객체와 질의 점 사이의 최단 거리가 짧은 순으로 변경한 후, 확장된 노드를 힙에 최단 경로 및 경로를 저장한다. 이와 같은 과정을 거쳐 생성한 최단 경로 및 거리는 도로 세그먼트를 확장할 때마다 모든 세그먼트를 검색할 필요가 없으며, 도로 세그먼트 정보를 디스크로부터 매번 액세스해야 하는 문제를 줄일 수 있다.

## 4. 성능평가

### 4.1 실험 데이터 및 환경

본 실험 수행을 위해, 실험 환경은 실제 도로 네트워크 데이터로 ORN (Oldenburg Road Network)와 CRN (California Road Network)를 사용한다[2]. 정적 객체

의 수는 각각의 도로 세그먼트 수 비율을 0.5%, 1%, 5%로 변화시켜 생성하였으며, 객체들의 위치를 임의로 분포시켜 설정하였다. 성능 평가를 위하여 다음과 같은 방법으로 수행하였다:

- INE: 모든 도로 세그먼트를 확장하면서 R\*-트리[8]를 이용하여 k개의 정적 객체를 찾는 방법이다[3].
- post-Dijkstra: k-NN 질의 처리 방법에 의해 k개의 정적 객체들이 주어졌을 때, Dijkstra 방법[7]을 이용한 방법이다.
- Naive-Astar: A\* 알고리즘[6]을 이용하여 개의 정적 객체에 대한 경로를 매번 반복 탐색하는 방법이다.
- Our-Method: 본 논문에서 제안한 방법으로, k-NN 질의 처리 방법에 의해 k개의 정적 객체에 대한 경로를 탐색하는 방법이다.

4.2 실험 결과 및 분석

그림 6은 정적 객체의 수 k를 10개로 고정하고, pseudo-k/k에 대해 pseudo-k=k(1+a)에서 a를 0.01, 0.15, 0.2, 0.25로 변경하여 수행 시간을 측정하였다. 그림을 보면, 제안한 Our Method는 기존 방법들에 비해 성능을 향상시킨 것으로 나타났다. 그러나, Our Method는 pseudo-k의 비율이 2% 이상인 경우, 성능이 post-Dijkstra 보다 저하된 것으로 나타났다. 이러한 원인은 pseudo-k의 수가 많을수록 검색해야 하는 객체의 수가 많기 때문에 차이를 보이는 것이며, pseudo-k의 비율이 1.5% 이상이면 높은 정확률과 함께 착오 기각 문제를 해결할 수 있다[2]. 따라서 Our Method는 다른 기법에 비해 성능이 우수한 것을 알 수 있다.

그림 7은 ORN과 CRN에 대해 도로 세그먼트에 따른 전체 정적 객체의 비율을 1%로 설정하고, 정적 객체의

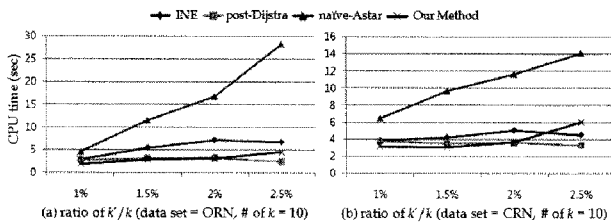


그림 6 pseudo-k/k의 비율 변화에 따른 검색 시간 비교

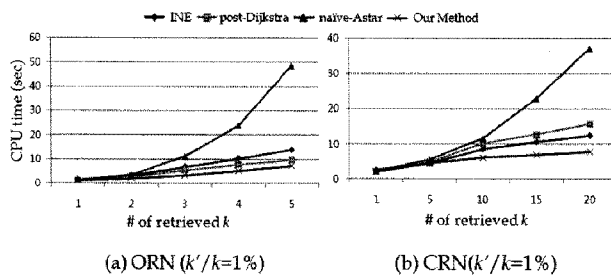


그림 7 객체 수의 변경에 대한 실험 비교

k를 달리하면서 수행 시간을 측정한 결과이다. 그림을 보면, k의 수가 증가할수록 Our Method이 기존 방법들 중에서 post-Dijkstra 보다 성능이 향상시켰음을 알 수 있다. 그 이유는 post-Dijkstra는 인접한 도로 세그먼트들을 모두 확장하는 반면에, Our Method는 인접한 도로 세그먼트를 모두 확장하는 불필요한 노드 계산을 줄였기 때문이다.

그림 8은 k-NN 질의에 대한 정적 객체 수는 10개로 고정하고, 도로 네트워크 내의 정적 객체의 수에 대하여 ORN과 CRN에서 각각 도로 세그먼트 수를 0.5%, 1%, 5%로 변경하면서 성능을 측정하였다. 그림을 보면, 모든 경우에 있어서 제안한 Our Method가 다른 방법에 비해 성능을 향상시켰음을 알 수 있다. 이는 앞서의 실험의 결과에서와 같이 Our Method가 모든 경우에 있어서 비교적 좋은 성능을 보이기 때문이다. 결국, 제안한 Our Method는 k의 수를 변경하는 경우나, 도로 세그먼트의 비율에 따른 경우에 있어서도 기존 방법에 비해 성능을 크게 향상시킨다고 말할 수 있다.

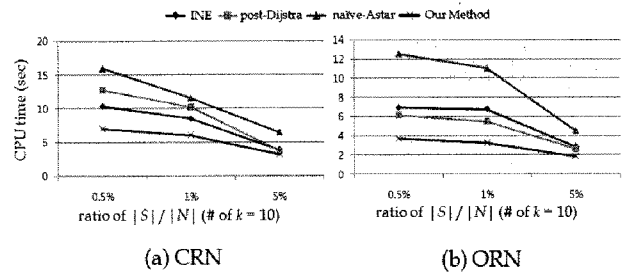


그림 8 |S|/|N|의 비율 변화(0.5%, 1%, 5%)에 따른 검색 시간 비교

5. 결론

도로 네트워크 상에서 k-NN 질의를 처리하기 위해서는 많은 비용이 요구된다. 이러한 질의를 처리하기 위해서 기존 연구에서는 다양한 방법들이 제시되었다. 하지만, 이들 방법은 모든 도로 세그먼트를 확장하면서 정적 객체들을 검색하거나, 최단의 경로 및 거리의 계산없이 정적 객체를 검색하는 문제점을 가지고 있다. 이에 본 논문에서는 k-NN 질의에 대해 최단 경로 탐색을 측정하는 방법을 제시하였다. 우선, k-NN 질의 처리 방법을 통하여 k개의 정적 객체를 찾고 각 정적 객체까지의 경로를 탐색한다. 탐색하는 과정에서 A\* 알고리즘을 개선하여 적용하기 때문에 효과적으로 질의 점으로부터 다수의 정적 객체까지의 경로를 탐색할 수 있다. 실험을 통하여 본 논문에서 제안하는 방법이 다른 방법에 비해 우수함을 보였다.

## 참고 문헌

- [1] Wu, S. and Wu, K., "Effective Location-Based Services with Dynamic Data Management in Mobile Environments," *Wireless Networks*, vol.12, no.3, pp.369-381, 2006.
- [2] Lee, S.-C., Kim, S.-W., Lee, J., and Yoo, J. S., "Approximate Indexing in Road Network Databases," *ACM Int'l Symp. on Applied Computing*, ACM SAC, pp.1568-1572, Mar. 2009.
- [3] Papadias, D., Zhang, J., Mamoulis, N., and Tao, Y., "Query Processing in Spatial Network Databases," In *Proc. Int'l Conf. on Very Large Data Bases*, VLDB, pp.802-813, Sept. 2003.
- [4] Kolahdouzan, M. and Shahabi, C., "Voronoi-Based K-Nearest Neighbor Search for Spatial Network Databases," In *Proc. Int'l Conf. on Very Large Data Bases*, VLDB, pp.840-851, Sept. 2004.
- [5] Faloutsos, C. and Lin, K., "FastMap: A Fast Algorithm for Indexing, Data-Mining and Visualization of Traditional and Multimedia Datasets," In *Proc. ACM Int'l Conf. on Management of Data*, ACM SIGMOD, pp.163-174, May 1995.
- [6] Hart, P. E., Nilsson, N. J., and Raphael, B., "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, pp.100-107, July 1968.
- [7] Dijkstra, E. W., "A note on two problems in connection with graphs," *Numerische Mathematik*, vol.1, pp.269-271, 1959.
- [8] Beckmann, N., Kriegel, H., Schneider, R., and Seeger, B., "The R\*-tree: An Efficient and Robust Access Method for Points and Rectangles," In *Proc. the ACM Int'l Conf. on Management of Data*, ACM SIGMOD, pp.322-331, May 1990.



신성현

2000년 관동대학교 컴퓨터공학과 학사  
 2002년 강원대학교 컴퓨터공학과 석사  
 2007년 강원대학교 컴퓨터공학과 박사  
 2000년~2007년 KAIST 첨단정보기술연구센터 연구보조원. 2007년~현재 한양대학교 BK21 사업단 정보기술분야 Post-Doc. 관심분야는 데이터 웨어하우스, OLAP, 데이터 마이닝, 이동 객체 데이터베이스, XML 데이터베이스 등



이상철

2005년 한양대학교 소프트웨어학과 학사  
 2007년 한양대학교 전자컴퓨터통신공학과 석사. 2007년~현재 한양대학교 전자컴퓨터통신공학과 박사과정. 관심분야는 데이터베이스 시스템, 데이터 마이닝, 공간 데이터베이스/GIS, 이동 객체 데이터베이스/텔레매틱스, 사회 연결망 분석

김상욱

정보과학회논문지 : 데이터베이스  
 제 36 권 제 1 호 참조



이정훈

1988년 서울대학교 컴퓨터공학과 학사  
 1990년 서울대학교 컴퓨터공학과 석사  
 1996년 서울대학교 컴퓨터공학과 박사  
 1997년~현재 제주대학교 전산통계학과 교수. 관심분야는 실시간 통신, 텔레매틱스 네트워크, 위치기반 서비스



임을규

1992년 서울대학교 컴퓨터공학과 학사  
 1994년 서울대학교 컴퓨터공학과 석사  
 2002년 University of Southern California 컴퓨터과학 박사. 2000년~2002년 WiseNut Inc. Sr. SW Engineer. 2002년~2005년 국가보안기술연구소 선임연구원. 2005년~현재 한양대학교 공과대학 컴퓨터공학부 조교수. 관심분야는 유무선 네트워크 보안, RFID 보안, 금융 보안, 병렬 프로그램