

# 3차원 지리 공간 환경에서의 근접주변 객체 질의 처리 기법

그로미코 디아나\*, 장홍준\*, 최우성\*, 현경석\*, 정재화\*\*, 정순영\*†

\*고려대학교 컴퓨터학과

\*\*한국방송통신대학교 컴퓨터학과

{dialen, hongjunjang, ws\_choi, ks\_hyun, jsy}@korea.ac.kr

jaehwachung@knou.ac.kr

## The Method of Nearest Surround Query Processing in Three-Dimensional Space

Diana Gromyko\*, Hong-Jun Jang\*, Woo-Sung Choi, Kyeong-Seok Hyun,\*  
Soon-Young Jung,\*† Jaehwa Chung\*\*

\*Dept. of Computer Science and Engineering, Korea University

\*\*Dept. of Computer Engineering, Korea National Open University

### 요 약

최근 모바일 기기의 발달과 기기의 보급으로 인하여, 사용자의 이동성에 따른 위치 정보에 관한 질의 요구가 증가되고 있다. 이러한 사회적 특성에 맞추어 현재 위치를 중심으로 주변 객체를 찾는 최근접주변 객체 질의에 대한 연구가 활발히 진행되고 있다. 기존 최근접 주변객체 질의[1]는 2차원 공간에서 질의 점 주변에 보이는 모든 객체를 검색하며 보이지 않는 객체는 제외시킨다. 하지만 3차원 공간 환경일 경우 객체들의 높이를 감안해야 하기 때문에 기존 기법들로는 3차원 근접주변 객체를 검색하기에는 적합하지 않다. 본 논문에서는 이런 문제점을 해결하는 새로운 3차원 최근접 주변객체 질의 (3dimensional Nearest Surround Query: 3dNS)를 제안한다. 객체들 간의 Z\_angle 수직각도를 사용함으로써 3차원 공간 환경에서 객체 검색을 가능하게 하며 질의 처리의 정확성을 높일 수 있다.

### 1. 서론

최근 모바일 기기의 발달과 보급으로 인하여 사용자의 이동성에 따른 위치 정보를 기반으로 하는 질의에 대한 요구가 증가되고 있다. 이러한 사회적 특성에 맞추어 현재 위치를 중심으로 주변 객체를 찾는 최근접 주변 객체 질의 NS[1,4]에 대한 연구가 활발히 진행되고 있다. 기존 NS 질의는 2차원 공간에서 질의 점 주변에 있는 보이는 모든 객체를 검색하며 보이지 않는 객체는 제외시킨다. 하지만 제외된 객체들이 3차원 공간 환경에서는 높이의 가시성 여부에 따라 NS 질의의 결과에 포함되는 경우가 있다. 그러므로 3차원 공간 환경에서는 객체를 검색할 수 있는 새로운 질의가 필요하다. 따라서 본 연구는 3차원 최근접 주변객체 질의(3dNS)를 제안한다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 기존 관련 연구에 대해 살펴보고, 3장에서는 3차원 지리적 공간 환경에서 효과적으로 근접주변 객체 질의를 처리하는 알고리즘을 제안한다. 4장은 실험결과에 대해 알아보고 마지막으로 5장에서 결론을 맺도록 하겠다.

### 2. 관련 연구

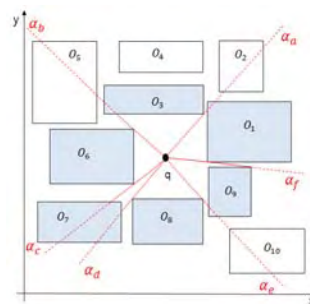
지금까지 근접주변 객체 질의에 대한 다양한 연구들이 진행되어 왔다. 그 중 가시 최근접(Visible Nearest

Neighbor: VNN) 질의[2,3,5,6,7]와 최근접 주변객체(Nearest Surround: NS) 질의[1,4,8,9]는 많은 연구가 이루어졌다.

VNN 질의는 질의 점으로부터 보이는 객체 중 가장 가까운 객체를 검색한다.

NS 질의는 질의 점을 중심으로 시계 반대 방향으로 주어진 각도 내에 보이는 객체들을 검색한다. NS 질의의 효율적인 처리를 위해 R-Tree[10]의 색인 구조를 사용한다.

(그림 1)은 NS 질의의 예이다.



(그림 1) NS 질의 예

검색된 NS 질의의 결과는  $\{O_1, [\alpha_f, \alpha_a], O_3, [\alpha_w, \alpha_b], O_6, [\alpha_b, \alpha_c], O_7, [\alpha_c, \alpha_d], O_8, [\alpha_d, \alpha_e], O_9, [\alpha_e, \alpha_f]\}$ 가 된다. 이와

같은 NS 질의의 결과는 경계 각도 내에서 질의 점과 객체의 최소 거리(min\_madist)와 최대 거리(max\_madist)를 계산하여 NS 질의의 결과를 결정하게 된다.

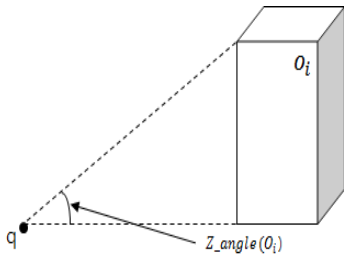
min\_madist는 질의 점과 객체 사이의 최단거리를 나타내고, max\_madist는 가까운 거리 중 가장 먼 거리를 나타낸다. (그림 1)에서  $O_2, O_4, O_5, O_{10}$  처럼 경계 각도 내에서 NS 결과 내에 포함되는 객체들에 의해 가려져 제외된 객체들이 3차원 공간 환경에서는 높이의 가시성 여부에 따라 NS 질의의 결과에 포함되는 경우가 있다. 이를 위해 본 논문에서는 3차원 공간 환경에서도 객체를 검색할 수 있는 3dNS 질의를 제안한다.

### 3. 3차원 근접 주변 객체 질의 처리

이 장에서는 3dNS 질의 처리를 위해 필요한 질의 점과 객체의 주변 각도, 유클리드 거리 그리고 객체들의 높이를 비교할 수 있는 수직각도 Z-angle 및 제거 규칙을 정의하고 이를 토대로 하는 3dNS 질의 처리 알고리즘을 제안한다.

#### 3.1 Z\_angle 정의

3dNS 질의 처리를 위해서 객체들 간의 높이를 비교할 수 있는 새로운 수직각도 필요하다. Z\_angle 수직각도는 다음과 같이 정의된다.



(그림 2) Z\_angle 정의

**정의 1. Z\_angle:** 질의 점 q와 집합 객체 O가 정해졌을 때, 수직각도 Z\_angle은 다음과 같이 정의한다

$$Z\_angle(O_i) = \arctan\left(\frac{O_i^h}{O_{i\max\_madist}}\right)$$

$O_i \in O$  포함 된 경우,  $O_i^h$ 는 객체의 높이와,  $O_{i\max\_madist}$ 는 객체의 max\_madist<sup>1)</sup>(q,  $O_i$ ,  $[\theta_o^+, \theta_o^-]$ )를 의미한다 (그림 2).

#### 3.2 3dNS 질의

본 논문에서는 3dNS 질의는 3차원 공간 환경에서 질의 점 주변에 보이는 모든 객체를 검색하는 질의이다.

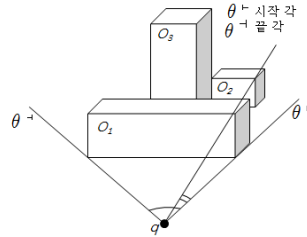
(그림 3)은 3dNS 질의의 예를 보여준다. 객체들의 집합  $O = \{O_1, O_2, O_3\}$ 가 주어진 상황에서 질의 점(q)과 수평각도( $\theta^+, \theta^-$ )가 주어졌을 때,  $O_1$ 이 q와 가장 가까운 객체인지만 이들의 높이를 비교하여 3dNS 질의의 결과는  $3dNS = \{O_1, O_3\}$ 가 된다.

객체의 집합 O에 대한 3dNS 질의는 다음과 같이 정의된다.

- 1) max\_madist는 수평각도 내에서 가까운 거리 중 가장 먼 거리
- 2)  $\theta_o^+$  - 시작 각,  $\theta_o^-$  - 끝 각

**정의 2. 3dNS 질의:** 질의 점 q와 집합 객체 O가 주어질 때, 3차원 공간 환경에서의 근접주변 객체 3dNS(q) 질의의 결과가 튜플  $\langle O: [\theta^+, \theta^-] \rangle$ 이라면, 3dNS(q) 질의의 결과는 아래와 같은 조건을 만족해야만 한다.

- 1)  $3dNS(q) = \{ \langle o: [\theta^+, \theta^-] \rangle \mid o \in O \wedge [\theta^+, \theta^-] \subseteq [0, 2\pi] \wedge \forall o' \in (O - \{o\}) \}$
- 2)  $\min\_madist^3(q, o, [\theta_o^+, \theta_o^-]) < \min\_madist(q, o', [\theta_{o'}^+, \theta_{o'}^-])$
- 3)  $Z\_angle(o) < Z\_angle(o')$

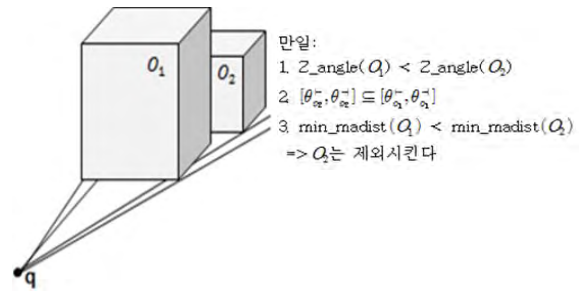


(그림 3) 3dNS 질의 예

#### 3.3 3dNS 질의 처리 성능 향상 방안

본 논문에서는 3dNS 질의 처리를 위해 R-Tree[10] 상에서 주어진 수평각도에 포함되지 않는 데이터를 질의 처리 연산에서 미리 제외시킨다.

**제거 규칙 1:** 질의 점 q와 집합 객체 O가 주어질 때, 만일  $Z\_angle(o_i) < Z\_angle(o_j)$ ,  $[\theta_{o_j}^+, \theta_{o_j}^-] \subseteq [\theta_{o_i}^+, \theta_{o_i}^-]$  하고,  $\min\_madist(o_i) < \min\_madist(o_j)$  라면  $o_j$ 는 제외시킨다.



(그림 4) 제거 규칙 1 적용 예

(그림 4)을 제거 규칙 1을 이용해 3dNS 질의의 결과를 살펴보면 다음과 같다.

먼저 질의 점 q에서부터 객체까지 가장 가까운 각의 범위를 찾아서 각 범위 내에 공통으로 들어가 있는 객체를 찾아보면  $O_1$ 과  $O_2$ 가 검색된다.  $O_1$ 과  $O_2$ 의 min\_madist를 비교해 보면  $O_1$ 의 min\_madist가  $O_2$ 의 min\_madist보다 작다는 ( $\min\_madist(O_1) < \min\_madist(O_2)$ ) 것을 알 수 있다. 그리고 추가된 수직각도 Z\_angle을 이용해 비교해보면  $O_2$ 의 Z\_angle이  $O_1$ 의 Z\_angle보다 크다. 그러므로  $O_2$ 가 질의 점

- 3) min\_madist는 수평각도 내에서 질의 점과 객체 사이 최단거리

q로부터 객체  $O_1$ 보다 멀리 떨어져 있으며,  $O_1$ 의 Z\_angle 보다 크고 가려져 있기 때문에 제거 규칙 1의 조건으로 보면  $O_2$ 는 제외시킨다. 즉 3dNS 질의의 결과는  $O_1$ 이 된다.

### 3.4 3dNS 질의 처리 알고리즘

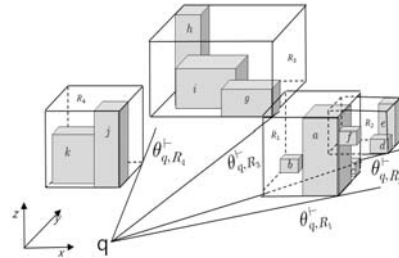
이 장에서는 기존 연구[1]인 NS질의 처리 알고리즘과 3dNS를 찾기 위해 추가된 수직각도 Z\_angle을 이용하는 3dNS 질의 처리 알고리즘을 살펴보겠다.

알고리즘 1을 이용해서 연구 예제 (그림 5)의 3dNS 검색 결과에 대해 살펴보도록 하겠다.

질의 점 q와 R-Tree에 의해 인덱스 된 4개의 부모 노드와 부모 노드가 보호하고 있는 11개의 리프 노드로 나누어져 있다.

먼저 3dNS를 검색하기 위해서는 질의 점 q를 둘러싸고 있는 객체들의 검색순위를 정해야 한다. 각각의 선은 검색 공간에서 객체들의 시작점을 나타내고, X축을 시작 점 0으로 해서 시계반대 방향 순으로 객체순위를 검색해야 한다. 객체순위를 정하기 위해서는 알고리즘 1의 6번째 줄에 있는 순위 함수를 이용해 검색순위를 정렬해보면  $[R_1, R_2, R_3, R_4]$ 와 같은 순으로 검색이 된다.

위와 같이 정렬된 검색순위를 순차적으로 삽입하면 a가 오게 되는데, 질의 점 q에서 X 축을 중심으로 검색하면 ( $0^\circ \sim \theta_a^+ : 0, \theta_a^+ \sim \theta_a^- : a, \theta_a^- \sim 2\pi : 0$ )와 같이 나타나는 것을 알 수 있다. 그리고 a가 리프 노드라서 알고리즘 1의 11번째 줄의 NS 삽입 함수를 이용해 검색 한 결과  $\langle a : [\theta_{q,a}^+, \theta_{q,a}^-] \rangle$ 를 토대로 3dNS입을 알 수 있다.



(그림 5) 3dNS 질의 처리 예

그 다음으로 객체 c를 삽입하게 되는데, c의 수평각도  $[\theta_{q,a}^+, \theta_{q,a}^-]$ 안에 포함되어 있으므로, 객체 a와 같은 공통 각 범위 내에 있는 것을 알 수 있다. 그러므로 알고리즘 1의 7번째 줄로 넘어가서 7번째 줄의 조건을 수행하게 되는데, c의 min\_madist가 a의 max\_madist보다 크고, c의 Z\_angle이 a의 Z\_angle 보다 크기 때문에 c를 삭제시켜야 한다.

순차적으로 부모  $R_2$ 를 알고리즘에 적용해보면 다음과 같다.  $R_2$ 가 객체 c와 같은 경우이기 때문에 삭제시키면 된다. 그리고  $R_2$ 가 부모 노드라서 리프 노드도 같이 삭제시킬 수 있다.

그 다음으로는 객체 b를 삽입시켜 보자. 객체 b는  $R_1$ 의 리프 노드라서 알고리즘 1의 11번째 줄의 NS 삽입 함수를 이용해 검색하게 된다. 이처럼 검색한 각 범위가  $\langle b : [\theta_{q,b}^+, \theta_{q,b}^-] \rangle$ 이며, 이미 3dNS로 도출된 결과와도 공통으로 겹치지 않기 때문에 3dNS가 된다.

나머지 객체들도 위와 같은 절차를 통해 질의 결과에의 포함여부가 결정된다.

## 4. 성능 분석

### 4.1 객체 수에 따른 성능 분석

Computation cost(cpu time)와 I/O cost(Accessed Nodes) 관점에서 제안하는 3dNS 질의 처리 알고리즘의 성능을 분석하였다.

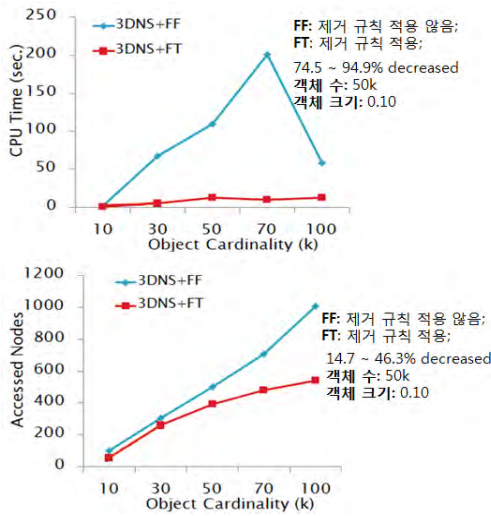
제거 규칙 적용하지 않는 3dNS 질의 처리 알고리즘과 비교하여 3dNS 질의 처리 알고리즘의 성능을 분석하였다. 제거 규칙이 적용되는 경우의 3dNS 질의 처리 알고리즘의 성능 비교하였다. 즉 제안하는 제거 규칙 1의 효과성을 분석하고자 하였다.

먼저 객체 수의 영향력을 평가하기 위하여 객체의 수를 만개에서 10만개까지 증가시켜 시간과 접근한 인덱스 노드의 수를 측정하였다. 즉, 객체수의 증가에 비례하여 성능은 감소하고 있음을 보여주고 있다. 그러나 객체수가 7만개 이상인 경우, 성능이 향상되고 있는데 이는 단위 면적당 객체 수 즉, 밀도가 커지는 경우 성능이 향상됨을 보

알고리즘1: 3dNS 질의 처리
<b>Input:</b> query point q, R-tree root index node
<b>Output:</b> 3dNS result
1. Let Q be priority queue;
2. Let $\alpha$ be angular range of examined objects, set $[0, \alpha]$ ;
3. Let N be the 3dNS result and be initialized as $\langle \perp : [0, 2\pi] \rangle$
4. while (Q is not empty) do
5. Let $\epsilon$ be an element (node or object);
6. $\epsilon \leftarrow$ Priority Q (q, Q, $[0, \alpha]$ );
7. if $\min\_madist(q, \epsilon, [\theta^+_{q, \epsilon}, \theta^-_{q, \epsilon}]) > \text{maximum } \max\_madist \wedge [\theta^+_{q, \epsilon}, \theta^-_{q, \epsilon}] \subseteq [0, \alpha] \wedge Z\_angle(\epsilon) \text{ mast}$
be more larger then $\epsilon$ can be skip;
8. else
9. if ( $\epsilon$ is node) then put all child entries to Q;
10. else
11. $N \leftarrow$ 3dNS;
12. $\alpha \leftarrow \max(\alpha, \theta^-_{q, \epsilon})$ ;
13. Output N

검색된 순위 객체가 부모 노드 또는 리프 노드라도 나타날 수 있다. 이와 같은 경우는 알고리즘 1의 9번째 줄에 있는 것처럼, 만일 검색순위의 객체가 부모 노드라면 리프 노드들 역시 검색순위로 다시 정렬해 주어야 한다. 이를 순차적으로 검색하게 되는데,  $R_1$ 이 부모 노드라서 리프 노드(a, b, c)를 삽입해서 객체순위를 다시 검색하면  $[a, c, R_2, b, R_3, R_4]$ 와 같이 나타나게 된다. 이처럼 나타나게 된 이유는 최소 거리로 객체순위를 정렬하기 때문이다.

여 준다 (그림 6).



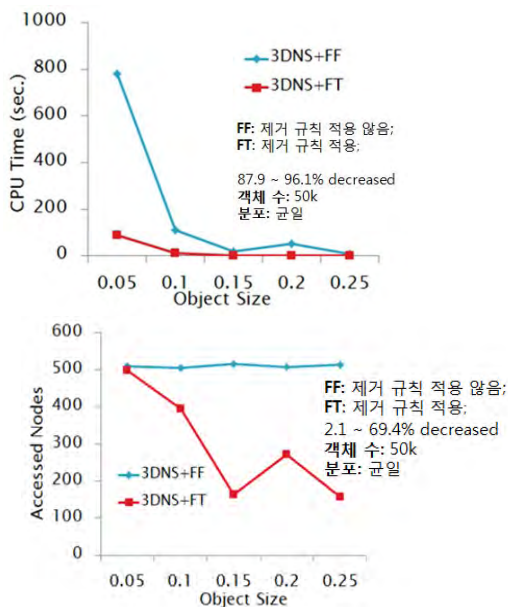
(그림 6) 객체 수의 영향력

#### 4.2 객체 크기에 따른 성능

각 실험에서 객체 수, 객체 크기, 객체 분포를 변인으로 적용하여 3dNS 검색 알고리즘과 각 제거 규칙의 적용에 따른 성능 변화를 분석하였다. Uniform 하게 분포된 질의 50개를 처리한 후 cpu time와 Accessed Nodes의 평균값을 측정하였다 (표 1).

(표 1) 실험 환경

파라미터	설정값
객체수	10k, 30k, 50k, 70k, 100k
객체크기	0.05, 0.10, 0.15, 0.20, 0.25
객체분포	Uniform, Gaussian(m.=500, $\sigma=250$ )
질의수	50(Uniform)



(그림 7) 객체 크기의 영향력

(그림 7)은 객체 크기에 따른 3dNS 검색 알고리즘과 각 제거 규칙의 적용에 따른 성능변화를 분석한 것이다.

결과는 객체 크기에 CPU cost와 Accessed Nodes 모두 반비례하고 있음을 보여주고 있다. 이는 객체의 크기가 클수록 제거 규칙이 적용될 수 있는 확률이 높아지며 이 뿐만 아니라 공간에 대한 객체의 밀도가 높아져 성능이 향상되는 것으로 해석되어 진다.

#### 5. 결론

본 논문에서 제안하는 방법은 유클리드 거리 및 각도 그리고 높이를 고려해 공통 각 범위 내에 있는 후보데이터들을 제거 규칙과 Z\_angle을 이용해 연산 단계 및 2차원에서 가려져 보이지 않았던 후보데이터까지 효과적으로 결과 객체로 검색할 수 있는 3차원 공간 환경에서의 근접 주변 객체 질의 처리 기법이다.

향후 연구 과제로 사용자가 이동함에 따라 발생하는 객체의 연속적인 질의 처리 기법과 모바일 환경에서 근접 주변 객체를 검색하여 질의 결과의 변화를 실시간으로 처리하는 연구가 필요하다

#### 참고문헌

- [1] K. C. K. Lee, W. C. Lee, and H. V. Leong, "Nearest Surround Queries", IEEE, pp. 1444 - 1458, 2010
- [2] Y. Wang, Y. Gao, L. Chen and Q. Li, "All Visible k Nearest Neighbor Queries", DEXA, pp. 392-407, 2012
- [3] S. Nutanong, E. Tanin and R. Zhang, "Incremental Evaluation of Visible Nearest Neighbor Queries", IEEE, pp. 665-681, 2010
- [4] X. Guo, B. Zheng, Y. Ishikawa, Y. Gao, "Direction based surrounder queries for mobile recommendations", VLDB, pp. 743-766, 2011
- [5] H. Xu, Z. Li, Y. Lu, K. Deng and X. Zhou, "Group Visible Nearest Neighbor Queries in Spatial Databases", WAIM, pp. 333-344, 2010
- [6] Y. Gao, B. Zheng, G. Chen, Q. Li, and X. Guo, "Continuous Visible Nearest Neighbor Query processing in Spatial databases," VLDB, pp. 371-396, 2011
- [7] S. Nutanong, E. Tanin, and R. Zhang, "Visible Nearest Neighbor Queries," DASFAA, pp. 876-883, 2007
- [8] K. C. K. Lee, J. Schiffman, B. Zheng, W.-C. Lee, and H. V. Leong, "Tracking Nearest Surrounders in Moving Object Environments," ICPS, pp. 3-12, 2006
- [9] K. C. K. Lee, J. Schiffman, B. Zheng, W.-C. Lee, and H. V. Leong, "Round-Eye: A system for tracking nearest surrounders in moving object environments," JSS, vol.80, issue 12, pp. 2063-2076, 2007
- [10] A. Guttman. "R-trees: a dynamic index structure for spatial searching", SIGMOD, pp. 47 - 57, 1984