행렬탐색을 이용한 양방향 각도제한 근접 점 계산방법

위영철 김하진
아주대학교 정보 및 컴퓨터공학부
{ycwee, hjkimn}@madang.ajou.ac.kr

Computing the Symmetric Angle Restricted Nearest Neighbors Using the Monotone Matrix Searching

Youngcheul Wee Hajin Kimn
Division of Information and Computer Engineering, Ajou University

요약

이 논문은 행렬탐색 방법을 이용하여 평면상의 n 개의 점에 대한 L_p, $1 \leq p \leq \infty$ 거리의 양방향 각도제한 근접 점 문제를 $\Theta(n \log n)$ 시간에 계산하는 알고리즘을 고안한다. 이 방법은 최적의 시간 복잡도를 가지며 궤적추적 법을 쓰지 않기 때문에 구현이 용이하고 실용적이다.

1. 서론

평면 상의 n 개의 점 $S=\{p_1, \ldots, p_n\}$에 대하여 한 점 p_i를 기준으로 평면을 k 개의 섹터로 나누었을 때 p_i의 한 섹터에 대한 각도제한 근접 점 (*angle restricted nearest neighbor: ARNN*)은 그 섹터에 있는 S의 점들 중에서 p_i에 가장 가까운 점이 된다 (그림 1 참조). Yao [8]에 의하여 소개된 각도제한 근접 점은 여러 가지 근접성 문제에 활용되었다. 예를들면, minimum spanning tree (MST) [8], relative neighborhood graph (RNG) [5], rectilinear Steiner tree [6], 최단거리 운동계획 [3] 등이 있다. Yao [8]는 d차원 공간의 n 개의 점에 대한 ARNN을 계산하는 $O(n^{2-a(d)} \log^{1-a(d)} n)$, $a(d)=2^{-(d+1)}$ 시간의 알고리즘을 소개하였다. Guibas와 Stolfi [4]는 L_1 and L_∞ 거리에서 평면상의 ARNN을 $O(n \log n)$ 시간에 계산하는 알고리즘을 고안하였다. Wee et al. [7]는 모든 L_p, $p \geq 1$ 거리에서 평면상의 ARNN을 계산하는 $O(n \log^2 n)$ 시간의 알고리즘을 소개하였다.

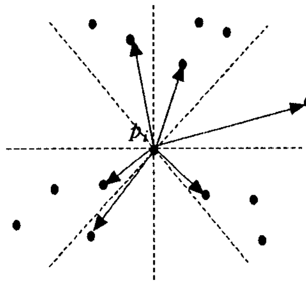

그림 1: p_i의 팔분면 각도제한 근접 점.

점 p_i의 각도제한 근접 점이 p_j 이고, 또한 점 p_j의 각도제한 근접 점이 p_i 이면 p_i 와 p_j는 서로 양방향 각도제한 근접 점이 된다. S에 대한 모든 양방향 각도제한 근접 점들을 에지로 연결한 그래프를 S의 양방향 근접 그래프라 하고 $SNN_k(S)$로 표

기한다. $SNN_8(S)$ 또한 모든 L_p 거리에서 $MST(S) \subseteq RNG(S) \subseteq SNN_8(S)$의 포함 관계를 만족하므로 대부분의 ARNN이 응용되는 문제들은 양방향 (symmetric) ARNN (SARNN)에 의하여 해결될 수 있다. 본 논문은 모든 L_p, $1 \le p$ 거리에서 $SNN_k(S)$을 계산하는 $\Theta(n \log n)$ 시간의 알고리즘을 소개한다. 본 방법은 divide and conquer 방법을 사용하며 merge 단계는 단조행렬 탐색방법 [1]을 이용한다. $SNN_8(S)$을 계산하는 방법을 일부 변형하여 모든 $k \ge 0$ 에 대한 $SNN_k(S)$를 계산 할 수 있으므로 논의는 팔분면 근접 그래프 계산에 국한한다. 남서서 북동동 양방향 근접점 그래프를 $ENWS(S)$로 나타내기로 하자. 본 방법의 기본구조는 먼저 S를 기울기가 45° 인 직선 $\overline{l}$을 기준으로 U와 V로 이분하여 $ENWS(U)$와 $ENWS(V)$를 재귀적으로 구축한 다음 U와 V간에 있는 양방향 근접 점 계산을 단조행렬 탐색 문제로 변환하여 Aggarwal 등의 [1] 단조행렬 탐색 알고리즘으로 $O(n)$ 시간에 $ENWS(U)$와 $ENWS(V)$를 merge 한다.

2. 팔분면 근접 점의 기하학적 특성

U와 V간의 양방향 근접 점 계산을 단조행렬 각 열의 최소 항을 찾는 문제로 변환하기 위해서 다음의 정리들이 필요하다. 점 b가 점 a의 남서서 팔분면에 놓여 있으면 b는 a에 지배 (dominate) 된다고 하자.

정리 2.1: $u_j \in U$가 $u_i \in U$를 지배하고 $v_k \in V$의 y 좌표가 u_i의 y 좌표 보다 같거나 작으면 v_k는 u_j와 양방향 팔분면 근접 점을 만들 수 없다.

증명: 그림 2 참조. □

정리 2.2: L_1 거리에서 S에 대한 한 점 $u_j \in U$의 남서서 팔분면 근접점이 U에 속하면 $u_j \in U$의 L_2 거리에서의 남서서 팔분면 근접점도 U에 속하게 된다.

증명: 그림 2 참조. □

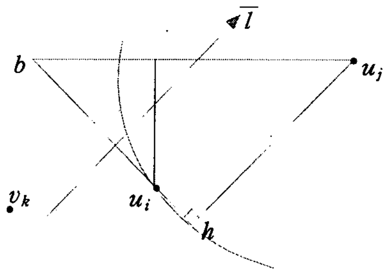

그림 2: 정리 2.1, 2.2의 증명 예.

정리 2.3: 세 점 $u_i, u_j, u_l \in U$에 대하여 u_j가 u_i를 지배하고 u_l의 y 좌표가 u_j의 y 좌표 보다 크거나 같고 한 점 $v_k \in V$의 y 좌표가 u_i의 y 좌표 보다 작거나 같으면 v_k는 u_l과 양방향 팔분면 근접 점을 형성할 수 없다.

증명: 정리 2.2와 유사함. □

3. 알고리즘

네 점 p, q, r, s가 사각형의 네 꼭지 점상에 위치하면 두 대각선의 길이 합은 마주보는 양변의 길이 합 보다 항상 크므로 단조행렬의 조건인

$d(p,r) + d(q,s) \ge d(p,q) + d(r,s)$ … 식(1)

의 관계가 성립하게된다. $u_j \in U$가 $u_i \in U$를 지배하지 않고 $v_l \in V$이 $v_k \in V$를 지배하지 않으면 네 점 u_j, u_i, v_k, v_l은 시계방향으로 한 사각형의 네 꼭지점이 되므로 식(1)이 성립한다. 따라서, $R \subseteq U$의 점들이 서로 지배되지 않고 $L \subseteq V$의 점들이 서로 지배되

지 않으면 R의 점들과 L의 점들과의 거리를 나타내는 행렬은 단조행렬 조건을 만족한다. U와 V를 정리 2.1과 정리 2.3에 따라서 $U=\{R_1, \dots R_t\}$와 $V=\{L_1, \dots L_t\}$로 분할하면, R_i의 점들과, L_j의 점들은 $i=j$일 때 만 서로 양방향 근접 점이 될 수 있다. U와 V간의 거리를 나타내는 행렬 M은 그림 3과 같이 왼쪽 하단과 오른쪽 상단이 계단모양의 ∞항을 나타내고 식(1)을 만족 하게되어 단조행렬이 된다.

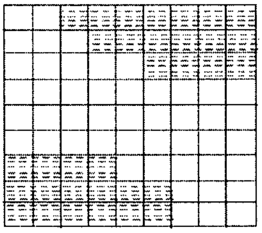

그림 3: $M=\{m(i,k)\}$의 구조.

위의 논의에 따라 M이 단조행렬이므로 단조행렬 탐색 알고리즘 [1]으로 $O(n)$ 시간에 U와 V간의 양방향 근접 점을 계산할 수 있으며 전체 divide and conquer 알고리즘의 소요시간은 $O(n\log n)$이 된다.

4. 결론

본 논문은 평면상의 n 개의 점에 대한 L_p, $1 \le p \le \infty$ 거리의 양방향 각도제한 근접 점 문제를 $\Theta(n\log n)$ 시간에 계산하는 알고리즘을 고안하였다. 이 방법은 최적의 시간 복잡도를 가지며 궤적추적 법을 쓰지 않기 때문에 구현이 용이하고 실용적이다.

본 논문과 연관된 미해결 문제들은

(1). Can $ARNN(S)$ under L_p metric, $1 < p < \infty$ be solved in $O(n\log n)$ time.

(2) Can $RNG(S)$ be computed from $SNN_8(S)$ in $O(n\log n)$ time [5]

등이 있다..

5. 참고문헌

[1] A. Aggarwal, M. M. Klawe, S. Moran, P. Shor, and R. Wilber, "Geometric Applications of a Matrix-Searching Algorithm", *Algorithmica* vol. 2, 1987, pp. 195-208.

[2] M. Ben-Or, "Lower Bounds for Algebraic Computation Trees", *Proc. of 15th Annual ACM Symposium on Theory of Computing*, 1983, pp. 80-86.

[3] K. Clarkson, "Approximation Problems for Shortest Path Motion Planning", *Proc. 19th Ann. ACM Symp. Theory of Computing,* 1987, pp. 56-65.

[4] L. J. Guibas and J. Stolfi, "On Computing all North-east Nearest Neighbors in the L_p metric", *Info. Process. Lett.* Vol. 17, 1983, pp. 219-223.

[5] J. Katajainen, "The Region Neighborhood Graphs, in the L_p metric", *Computing* Vol. 40, 1988, pp. 147-161.

[6] Y. C. Wee and S. Chaiken and S. S. Ravi, "Rectilinear Steiner Tree Heuristics and Minimum Spanning Tree Algorithms Using Geographic Nearest Neighbors", *Algorithmica* 12, 1994, pp. 421-435.

[7] Y. C. Wee, S. Chaiken and D. E. Willard, "On the Angle Restricted Nearest Neighbor Problem", *Info. Process. Lett.* Vol 34. 1990, pp. 71-76.

[8] A. C. Yao, "On constructing Minimum Spanning Trees in k-dimensional Spaces and Related Problems", *SIAM J. on Computing*, Vol. 11, 1982, pp. 721-736.