

제한된 분지수를 가지는 최소지름 신장트리 (A Minimum-diameter Spanning Tree with Bounded Degrees)

안 희 갑 * 신 찬 수 **
(Heekap Ahn) (Chan-Su Shin)

요 약 이차원 평면에 주어진 n 개의 점을 연결하는 신장트리(spanning tree) 중에서, 지름이 최소가 되는 트리는 특정 점에서의 분지수(degree)가 $n-1$ 까지 증가할 수 있다. 신장트리가 실제 네트워크 구조로 사용된다면 높은 분지수를 갖는 노드에선 작업 집중현상이 발생하여 전체 네트워크의 성능을 저하시킬 수 있다. 따라서 작은 분지수와 작은 지름을 갖는 트리를 구성하는 것은 중요하다. 본 논문에서는 (1) 트리의 분지수를 자유롭게 조절할 수 있고, (2) 트리의 지름이 최소 지름보다 상수 배 이상 크지 않고, (3) 임의의 점을 루트로 정하여 트리를 구하더라도 항상 단조(monotone)하며, (4) 트리의 에지들이 서로 교차하지 않는 신장트리 구성 알고리즘을 제안한다. 여기서 트리가 단조하다는 것은 루트부터 시작하여 임의의 노드가 지 연결되는 경로위에 있는 점은 루트로부터의 유클리디언 거리가 단조 증가하는 것을 의미한다. 이 단조성은 신장트리를 가시화할 때의 중요한 미적 기준으로 사용될 수 있다.

키워드 : 계산기하학, 알고리즘 설계 및 분석, 트리 네트워크

Abstract Given a set S of n points in the plane, a minimum-diameter spanning tree(MDST) for the set might have a degree up to $n-1$. This might cause the degradation of the network performance because the node with high degree should handle much more requests than others relatively. Thus it is important to construct a spanning tree network with small degree and diameter. This paper presents an algorithm to construct a spanning tree for S satisfying the following four conditions: (1) the degree is controlled as an input, (2) the tree diameter is no more than constant times the diameter of MDST, (3) the tree is monotone (even if arbitrary point is fixed as a root of the tree) in the sense that the Euclidean distance from the root to any node on the path to any leaf node is not decreasing, and (4) there are no crossings between edges of the tree. The monotone property will play a role as an aesthetic criterion in visualizing the tree in the plane.

Key words : Computational Geometry, Algorithm Design and Analysis, Tree Networks

1. 서 론

이차원 평면에 주어진 n 개의 점들로 구성된 집합을 S 라 하자. 이 점들을 서로 교차하지 않는 선분으로 연결한 트리를 신장 트리(spanning tree)라 한다. 이 트리에서 각 에지의 길이는 에지 양 끝 점 사이의 유클리디언 거리로 정의된다. 트리에서 두 점을 연결한 경로(path)는 유일하게 정의되고, 경로의 길이는 경로를 구성하는 에지들의 길이의 합으로 정의된다. 트리에서 두 점 사이의 거리는 두 점을 연결하는 경로의 길이로 정

의된다. 트리에서 가장 먼 거리를 이루는 두 점 사이의 거리를 트리의 지름(diameter)이라고 한다.

최소 지름 신장 트리(MDST: Minimum-Diameter Spanning Tree)는 S 의 점을 연결하는 신장트리 가운데 지름이 가장 작은 트리이다. 각 점을 도시로 생각하고 여러 도시를 연결하는 트리 네트워크를 구성한다고 하자. 만약 트리 네트워크의 설치비용을 줄이고 싶다면, 네트워크 연결선의 총 길이를 줄여야 한다. 이것은 트리 네트워크 중에서 에지의 길이 합이 최소인 MST (Minimum Spanning Tree)로 트리 네트워크를 구성해야 함을 의미한다. 그러나 통신 서비스의 공평성을 더 중요하게 생각한다면 전체 에지의 길이의 합보다는 가장 멀리 떨어진 두 도시간의 통신 비용(즉 네트워크에서의 거리)을 더 줄여야 한다. 이 경우엔 네트워크의 지름이 최소가 되는 MDST를 이용하는 것이 바람직하다.

* 본 연구는 한국과학재단 목격기초연구 R05-2002-000-00780-0 지원으로 수행되었음

† 비 회 원 : 한국과학기술원 전산학과 교수
heekap@hanmail.net

** 종신회원 : 한국외국어대학교 디지털정보공학과 교수
cssin@hufs.ac.kr

논문접수 : 2003년 8월 1일
심사완료 : 2003년 11월 13일

점집합 S 에 대한 MDST를 계산하는 최적 시간 알고리즘은 알려져 있지 않다. 현재까지 알려진 가장 빠른 알고리즘[1]은 1991년에 발표된 $O(n^3)$ 시간 알고리즘이다. 수행 시간이 크기 때문에, 최소 지름과 매우 가까운 근사된(approximate) 트리를 계산하는 알고리즘[2]이 최근에 발표되었다. 이 알고리즘에서는 임의의 실수 $0 < \epsilon < 1$ 에 대해 S 의 MDST 지름보다 $(1 + \epsilon)$ 배 이내의 지름을 갖는 트리를 선형시간에 계산할 수 있음을 보였다.

Ho 등[1]에 의하면, S 에 대한 MDST는 단항 트리(monopolar tree)이거나 이항 트리(dipolar tree)임이 증명되었다. 단항 트리란 S 의 한 점이 단항 중심으로 정의되고 나머지 모든 점은 단항 중심에 연결된 형태의 트리이며, 이항 트리는 두 점이 이항 중심으로 정의되며, 두 중심은 서로 선분으로 연결되고 나머지 점들은 두 이항 중심 가운데 하나에 연결된 트리로 정의된다. 같은 저자들은 또한 최소 지름의 단항 트리는 $O(n \log n)$ 시간에, 이항 트리는 $O(n^3)$ 시간에 계산할 수 있음을 보여, 결국 $O(n^3)$ 시간에 S 에 대한 MDST의 구성이 가능함을 보였다.

만약 MDST가 기간 통신망으로 사용된다면, 각 노드에서 처리해야 할 작업의 양이 적절해야 한다. 이것은 신뢰성 높은 통신 서비스를 위해 필수적이다. 한 노드에서 처리해야 할 작업의 양은 노드에 연결된 통신 에지의 수와 밀접한 관계가 있기 때문에, 많은 에지가 노드에 연결되어 있다면 그만큼 많은 작업을 해당 노드에서 처리해야 할 가능성이 높아진다. 그런데 위에서 살펴봤듯이, MDST는 단항 트리 또는 이항 트리이므로, 한 노드에 연결된 에지 수는 $n-1$ 까지 증가할 수 있다. 트리의 각 점의 분지수(degree)를 그 점에 연결된 에지의 개수로 정의하고 트리의 분지수를 노드의 분지수중에서 가장 큰 분지수로 정의한다면, MDST의 분지수는 결국 $n-1$ 까지 증가할 수 있다. 반면에, MST에서는 모든 노드의 분지수가 5를 넘지 않는다[4].

트리의 특정 노드 v_0 를 루트 노드로 정하여 부모 노드와 자식 노드의 관계가 존재하는 트리로 변환했을 때, 루트 v_0 에서 임의의 리프 노드 v_k 까지의 경로 $P = \langle v_0, v_1, \dots, v_k \rangle$ 를 고려해보자. 두 점 p, q 사이의 유클리드 거리는 $|pq|$ 로 표기하고, 트리의 경로 P 의 길이는 $|P|$ 로 표기한다. 임의의 리프노드 v_k 와 모든 $0 < i < k$ 에 대해 $|v_0 v_i| \leq |v_0 v_{i+1}|$ 이라면 이 트리는 (루트 v_0 에 대해) 단조(monotone)하다고 말한다. 이 단조성이 네트워크 자체의 중요한 성질은 아니지만, 네트워크를 이차원 평면에 가시화하는 경우에는 네트워크의 구조를

정확하게 표현할 수 있는 미적 기준으로 사용할 수 있는 중요한 성질이다.

본 논문에서는 트리의 분지수를 자유롭게 조절할 수 있고, 트리의 지름은 MDST의 지름보다 상수 배 이상 차이가 나지 않으면서, 임의의 루트 노드에 대해 단조한 신장트리를 구성하는 문제를 살펴본다. 이와 유사한 문제에 대해서는 Arya 등[3]에 의해 연구된 바 있다. 이 논문에서는 트리의 분지수를 특정한 상수로 제한하며 트리의 지름은 최소 지름의 상수 배 이내로 보장하는 알고리즘을 제안했다. 그러나 트리의 단조 성질에 대해서는 전혀 고려하지 않았다. 실제로, 제시된 알고리즘의 트리 구성 알고리즘을 적용할 때, 어떤 점을 루트로 선택하더라도 단조한 트리를 만들 수 없는 점 집합 S 의 예가 존재한다. 그러나 본 논문에서 제시하는 알고리즘은 다음 네 가지 조건을 만족하는 알고리즘을 제시한다: (1) 트리의 에지는 서로 교차하지 않고, (2) 트리의 지름은 S 에 대한 MDST의 지름보다 상수 배를 넘지 않고, (3) 트리의 분지수는 임의의 자연수 $m (\geq 2)$ 에 대해 $3m-1$ 이 되고, (4) 트리는 S 의 어떤 점을 루트로 정하여 트리를 구성하더라도 항상 단조하다.

본 논문에서 제안하는 알고리즘이 구성한 트리는 Arya 등[3]의 알고리즘이 구성한 트리의 성질을 그대로 유지하면서 임의의 점을 루트 노드로 결정하더라도 항상 단조하다는 것과 파라미터 m 을 조절하여 트리의 분지수를 적당한 값으로 조절할 수 있다는 장점이 있다.

2. 분지수가 제한된 트리 구성 알고리즘

S 의 점 중에서 가장 멀리 떨어진 두 점 사이의 거리를 S 의 지름(diameter)라 부른다. 이 두 점을 x, y 라 하자. 편의상 직선 \overline{xy} 는 수직선이고 x 가 y 보다 위쪽에 있다고 가정하자. C_1 은 x 를 중심으로 y 를 지나는 반지름이 $D = |xy|$ 인 디스크(disk)라고 하고, C_2 은 y 를 중심으로 x 를 지나는 반지름이 $D = |xy|$ 인 디스크라 하자. 점 x 를 지나면서 \overline{xy} 에 수평인 직선은 C_1 을 두 개의 반쪽 디스크로 분할한다. 그 중에서 아래 반쪽 디스크를 C 라 하자. 그러면 $C_1 \cap C_2$ 은 S 의 모든 점을 포함하고 $(C_1 \cap C_2) \subseteq C$ 이므로 C 는 S 의 모든 점을 포함한다(그림 1(a) 참조).

S 의 지름을 결정하는 두 점 중에서 한 점 - 본 논문에서는 x - 을 트리의 루트로 정했을 때의 트리 구성 알고리즘을 먼저 설명한다. S 의 임의의 점을 루트로 하는 경우에는 같은 알고리즘을 그대로 적용할 수 있다. 구체적인 방법은 2장의 마지막 부분에서 설명된다.

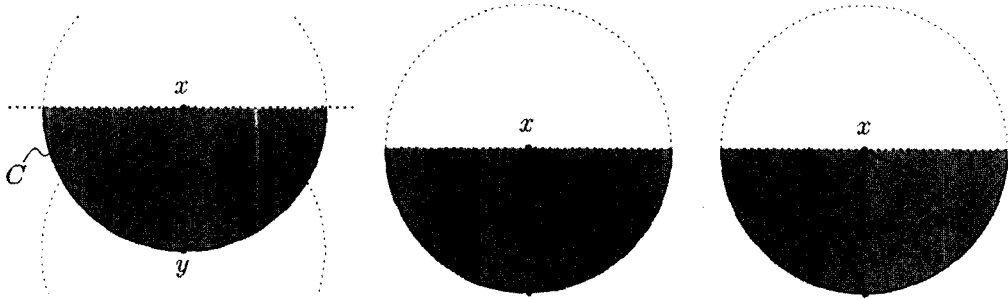


그림 1 (a) 반쪽 디스크 C와 첫 번째 레벨. (b) 두 번째 레벨. (c) 세 번째 레벨

2.1 트리 구성 알고리즘

S의 지름을 결정하는 두 점 중의 한 점 x 를 중심으로 반지름이 $D = |xy|$ 인 아래 반쪽 디스크 C는 그림 1(a)처럼 S의 모든 점들을 포함한다. 이 반쪽 디스크 C를 (x 를 중심으로) m 개의 같은 각의 부채꼴로 나눈다. 이렇게 나뉘어진 각 부채꼴 K 에 포함된 S의 점 중에서 한 점을 우선 선택한다. 이 점이 루트 x 에 연결되어 x 의 자식 노드 중 하나가 되며, 자신의 부채꼴에 포함된 다른 점들을 연결하여 구성될 부분트리(subtree)의 루트 노드의 역할을 한다. 이 점을 해당 영역의 루트라 부른다. 다음 단계에서는 각 부채꼴 K 는 다시 부채꼴과 유사한 형태의 영역(region) m 개로 나뉜다. 이 영역에 포함된 S의 점 중에서 하나 이상의 점을 선택하여 체인(chain)들로 연결하고 그 중에서 한 점을 해당 영역의 루트로 설정하고 부채꼴 K 의 루트 점과 연결한다. 이와 같은 방법으로 트리의 레벨(level)을 차례로 구성하여, S의 모든 점이 트리에 연결될 때까지 반복한다.

트리의 전체 루트가 첫 번째 레벨에 해당하며 루트의 자식 노드들이 두 번째 레벨을 구성하고 리프 노드로 내려갈수록 레벨이 증가하게 된다. 또한 알고리즘에서는 반쪽 디스크 C를 작은 부채꼴과 유사한 작은 영역들로 재귀적으로 분할하는데 이때 분할된 각 영역의 레벨은 그 영역의 루트의 레벨과 동일한 것으로 간주한다. 이제 현재 구성된 트리의 레벨에서 다음 레벨을 어떻게 구성하는지 차례로 살펴보자.

첫 번째와 두 번째 레벨: 트리의 첫 번째 레벨은 루트 노드만으로 구성되어 있으므로 점 x 를 루트로 설정하면 된다. 두 번째 레벨은 첫 번째 레벨의 노드에 연결되는 점들로 구성된다. 이를 위해, 반쪽 디스크 C를 x 를 중심으로 각이 π/m 가 되는 m 개의 부채꼴로 나눈다. (그림 1(b) 참조) 각 부채꼴 K 에 포함된 점들 중에서 x 로부터 가장 가까운 점 p 를 선택하여 x 에 선분으로 연결하면 두 번째 레벨이 완성된다. x 를 중심으로 점 p 를

지나는 디스크를 C_p 라 하자. 이제 점 p 는 부채꼴과 유사한 영역인 $\hat{K} = K - C_p$ 의 루트 역할을 하게 된다.

세 번째 레벨: 두 번째 레벨에서 정의된 각 부채꼴 K 에 대해, 점 p 가 선택되어 K 의 루트가 되었고 부채꼴과 유사한 영역인 \hat{K} 이 정의되었다. 세 번째 레벨에선 \hat{K} 를 다시 x 를 중심으로 같은 각을 갖는 m 개의 영역으로 나눈다(그림 1(c) 참조). 그 영역들을 왼쪽부터 $\hat{K}_1, \hat{K}_2, \dots, \hat{K}_m$ 이라 하자. 이제, 각 영역 \hat{K}_z 에 대해, \hat{K}_z 에 포함된 S의 점 집합 $S(\hat{K}_z)$ 이 공집합이 아니면 반드시 한 점 이상을 선택하여 체인으로 연결하고, 그 중에서 x 로부터 가장 가까운 점을 선택하여 p 와 연결한 후에 \hat{K}_z 의 루트로 설정한다. 이 과정을 자세히 살펴보자(그림 2도 함께 참조하자).

편의상 점 p 가 어떤 영역 \hat{K}_z ($1 \leq z \leq m$)의 윗변에 온다고 가정하고 점 집합 S_L 과 S_R 를 다음과 같이 정의한다: $S_L = S(\hat{K}_1) \cup \dots \cup S(\hat{K}_{z-1})$, $S_R = S(\hat{K}_{z+1}) \cup \dots \cup S(\hat{K}_m)$. 당연히 $S(\hat{K}) = S_L \cup S(\hat{K}_z) \cup S_R$ 이다.

지금부터 직선 \overline{px} 는 수직선이고, 점 p 는 윗변에 위치한다고 가정한다(그림 2 참조). 우선 점 p 가 위치한 영역 \hat{K}_z 에 대해, 전체 루트 x 에서 가장 가까운 $S(\hat{K}_z)$ 의 점을 선택해 p 의 자식노드로 연결한다. 이 점이 다음 레벨에서 \hat{K}_z 에 대한 루트가 된다. 다음으로 S_R 의 점 중에서 하나 이상의 점을 선택하여 체인으로 연결한다. 이 체인을 R -체인이라 부른다(S_L 에 대해서는 대칭적으로 L -체인을 정의할 수 있고, R -체인을 구성하는 것과 같은 방법으로 구성할 수 있다. 따라서 L -체인 구성 방법은 생략한다). 우선 p 와 K 의 오른쪽 위 꼭지점을 지나는 가상의 직선을 ℓ 이라 하자. 점 p 를 중심으로 ℓ 을 시계 방향으로 회전하면서 처음으로 만나는 S_R 의 점을 p_1 이라 하자. 다시 점 p_1 을 중심으로

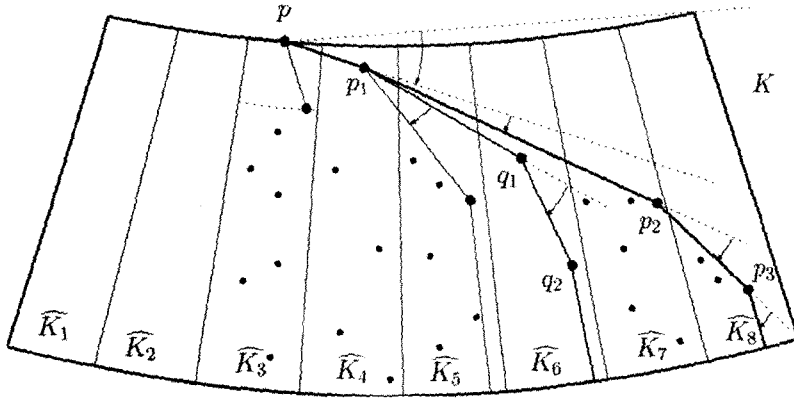


그림 2 세 번째 레벨의 구성

ℓ 을 계속 시계 방향으로 회전하면서 처음으로 만나서 S_R 의 점 p_2 를 찾는다. 이 방법을 계속해서 적용하여 p_3, p_4, \dots, p_i 를 찾는다. 이것은 ℓ 이 x 를 만날 때까지 반복한다. 점 p 와 p_1, p_2, \dots, p_i 를 차례대로 연결하면 위쪽으로 볼록한 R -체인(upward convex chain)이 된다. 이 볼록 체인의 윗부분에는 S_R 의 어떠한 점도 존재하지 않는다는 사실에 유의하자.

인접한 두 점 p_k 와 p_{k+1} ($1 \leq k < i$)에 대해, $p_k \in S(\widehat{K}_z)$ 이고 $p_{k+1} \in S(\widehat{K}_j)$ 이라고 하자. 여기서 $z < i < j \leq m$ 이다. 만약 $j-i=0$ 이라면(즉 $i=j$ 이라면), p_k 와 p_{k+1} 은 같은 영역 \widehat{K}_i (또는 \widehat{K}_j)에서 선택되어 R -체인으로 연결된 경우이다. 그러나 $j-i=1$ 이라면, 서로 인접한 영역에서 점 p_k 와 p_{k+1} 이 각각 하나씩 선택되어 R -체인에 참여한 것이다. 마지막으로 $j-i \geq 2$ 이라면, 영역 $\widehat{K}_{i+1}, \widehat{K}_{i+2}, \dots, \widehat{K}_{j-1}$ 에서는 어떠한 점도 선택이 되지 않았음을 의미한다(그림 2에서 \widehat{K}_5 와 \widehat{K}_6 가 이 경우에 해당하며 $p_k = p_1, p_{k+1} = p_2$ 가 된다). 이 영역들에 대한 루트를 결정해야 한다. 이를 위해, 점 p_k 에서 시작하는 새로운 R -체인을 $\widehat{K}_{i+1}, \widehat{K}_{i+2}, \dots, \widehat{K}_{j-1}$ 에 포함된 점들에 대해 위와 같은 방법을 재귀적으로 적용하여 구성한다.

$S' = S(\widehat{K}_{i+1}) \cup \dots \cup S(\widehat{K}_{j-1})$ 이라 하자. 우선 직선 ℓ 을 선분 $\overline{p_k p_{k+1}}$ 에 일치시킨 후에, p_k 를 중심으로 시계 방향으로 회전하면서 가장 처음 만나서 S' 의 한 점 q_1 을 찾고, 다시 q_1 을 중심으로 ℓ 을 회전시켜 처음 만나는 점 q_2 를 반복해서 찾는다. 이렇게 찾은 q_1, q_2, \dots, q_s 을 차례로 연결하고 q_1 을 p_k 에 연결하여 위로 볼록한 R -체인을 만든다(그림 2에서는 두 점

q_1, q_2 을 선택하여 점 p_1 과 함께 R -체인을 구성하였다). 물론 이 과정에서도 $\widehat{K}_{i+1}, \widehat{K}_{i+2}, \dots, \widehat{K}_{j-1}$ 의 어떤 영역의 점들은 R -체인이 포함하지 못할 수도 있다(그림 2에서는 \widehat{K}_5 의 어떤 점도 선택되지 않았다). 그렇다면 다시 재귀적인 방법을 적용하여 R -체인을 생성한다. 이것은 $\widehat{K}_z, \widehat{K}_{z+1}, \dots, \widehat{K}_m$ 의 각 영역에서 하나 이상의 점이 선택되어 R -체인에 연결될 때까지 반복된다. 그림 2에서는 네 개의 R -체인이 생성되었다. 어떤 영역에서는(그림에선 \widehat{K}_6) 두 점 이상이 선택될 수도 있지만, 그 점들은 반드시 동일한 R -체인으로 연결된다.

이젠 \widehat{K}_i ($z < i \leq m$)에서 R -체인으로 연결된 점 중에서 x 에 가까운 점(R -체인에서 보다 위쪽에 있는 점)을 \widehat{K}_i 의 루트 r_i 로 설정한다(사실 2 참조). 이 루트로부터 다음 레벨에서 선택된 점들이 연결된다(예를 들어 \widehat{K}_5 에서는 두 개의 점이 R -체인으로 연결되었고 그 중에서 x 에 가까운 q_1 이 \widehat{K}_5 의 루트로 결정되었다). 마지막으로 이 R -체인에 추가로 연결될 점들이 존재하는데 지금부터는 이 부분에 대해 자세히 살펴보자.

점 p 에서 시작하여 \widehat{K}_i 의 루트 r_i 를 지나서 R -체인 중에서 가장 아래쪽 체인을 R_i 라 하자. 이때 R_i 의 위쪽 영역을 R_i^+ 이라 하고 아래쪽 영역을 R_i^- 라 한다. 또한 점 r_i 를 지나면서 $\overline{p_k p_{k+1}}$ 에 수직인 직선 h_i 라 하고, h_i 에 의해 나누어지는 두 개의 반평면 중에서 x 를 포함하는 반평면을 h_i^+ 라 하고 다른 반평면을 h_i^- 라 하자. 그러면 \widehat{K}_i 에 대한 삼각주(delta)는 $\Delta_i = R_i^+ \cap \widehat{K}_i \cap h_i^+$ 로 정의한다. 그림 3에서 색칠된 부분이 삼각주이다. 만약, Δ_i 에 점들이 존재한다면, 즉 $S(\Delta_i) \neq \emptyset$ 이라면, 그

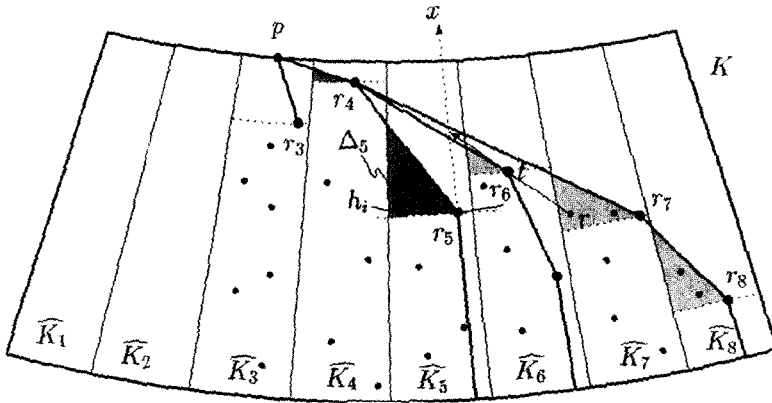


그림 3 세 번째 레벨에서의 삼각주 영역의 정의와 각 영역의 루트 설정

점들의 루트를 결정해야 하는데, 단조성질을 만족하기 위해선 r_i 가 아닌 다른 점을 루트로 선택해야 한다.

그림 3에서 Δ_5 와 Δ_7 을 비교해보자. 우선 Δ_5 에 있는 점들은 r_5 가 속한 R-체인에서 r_5 의 부모 점(여기선 점 r_4)을 루트 점으로 하여 재귀적으로 연결하면 된다. 물론, 이 경우엔 r_4 자체가 자신의 영역 \bar{K}_4 의 루트이기도 하므로 $S(\bar{K}_4)$ 과 $S(\Delta_5)$ 에 있는 점들 모두의 루트가 되어야 한다. 그러나 Δ_7 에 있는 점들은 상황이 다르다. R-체인 위에서 점 r_i 의 부모는 r_j 가 된다. 그런데 r_4 에서는 자신을 지나가는 R-체인과 함께 자신으로부터 시작하는 R-체인도 존재한다(만약, 자신으로부터 시작하는 R-체인이 두 개 이상이라면 그 중에서 가장 위쪽에 있는 R-체인만을 고려한다). x 에서 가장 가까운 $S(\Delta_i)$ 의 점을 r_i 라 하자.

이하의 설명을 위해 그림 3을 참조하라. 우선 r_4 에서 r_i 이 보이면 두 점을 선분을 연결한다. R-체인의 정의에 의해, 서로 인접한 두 R-체인 사이의 영역엔 어떠한 점도 존재하지 않기 때문에 이 선분은 트리의 다른 예지와 교차하지 않는다. 만약, 그림 3처럼 r_i 보이지 않는다면, 그것은 r_4 에서 시작하는 R-체인이 가로 막고 있기 때문이다. 그러면 r_4 에서 그 R-체인에 접선을 구해 점점 s 를 찾아 r_4 와 선분으로 연결한다. 이 선분 역시 다른 R-체인과 교차하지 않는다.

이젠 r_i 이 Δ_i 의 루트 점이 되어 다음 레벨에서 재귀적으로 같은 일을 반복한다. 여기서 증명해야 할 사실은 s 와 r_i 이 x 에 대해 단조하다는 것, 즉 $|xs| \leq |xr_i|$ 임을 보이는 것이다. 이것은 t 가 디스크 C 에 포함된다는 것을 증명하면 된다. 그런데 사실 1에서 증명될 성질에 의하

면, $|xp| \leq |xr_i|$ 이고 $|xp| \leq |xr_4|$ 이므로 선분 \overline{rx} , $\overline{r_4x}$ 과 $\overline{r_i x}$ 은 모두 C , 내부에 포함된다. 또한 t 가 r_i 이 보이지 않도록 r 을 가렸기 때문에 s 는 삼각형 $\Delta(x, r_4, r_i)$ 에 포함된다. 그런데 이 삼각형은 세 변 모두 C 에 포함되므로 $\Delta(x, r_4, r_i)$ 역시 C , 내부에 포함된다. 따라서 t 역시 C 에 포함된다. 결론적으로 삼각주에 대한 루트를 단조 성질을 유지하면서 정할 수 있다.

현재 레벨에서 영역 \bar{K}_i 에 루트 r_i 가 다음 레벨에서 책임져야 할 영역은 (1) r_i 가 점 p 가 위치한 영역 \bar{K}_2 의 루트라면 $\Delta_i = \bar{K}_i \cap C$,이 되고, 아니라면 (2-1) $\Delta_i = \bar{R}_i \cap \bar{K}_i \cap h_i$ 이 되거나 (2-2) (오른쪽 영역 \bar{K}_{i+1} 의 삼각주 Δ_{i+1} 도 함께 책임져야 한다면) 두 영역을 합병한 $\Delta_i \cup \Delta_{i+1}$ 영역이 되어야 한다. 마지막으로 (3) 삼각주 Δ_i 가 자신의 왼쪽 영역 \bar{K}_{i-1} 과 합병되지 않는다면, x 에서 가장 가까운 $S(\Delta_i)$ 의 점이 루트가 되어 Δ_i 를 책임진다.

네 번째 또는 그 보다 큰 레벨: 이 경우엔 세 번째 레벨과 거의 동일하다. 유일한 차이점은 네 번째(또는 그 보다 큰) 레벨의 루트가 책임질 영역의 모양이 조금씩 다른 것뿐이다.

2.2 구성된 트리의 성질 분석

위의 방법에 의해 얻어진 트리 T 의 단조성, 분지수, 그리고 지름을 분석해 보자.

단조성 분석: 트리 T 가 루트 x 에 대해 단조하다는 것을 증명하기 위해선 다음의 사실이 성립해야 한다.

사실 1. 임의의 레벨의 영역 K 와 K 의 루트 p 를 고려하자. K 에 포함된 임의의 점 a 에 대해, $|xp| \leq |xa|$ 이

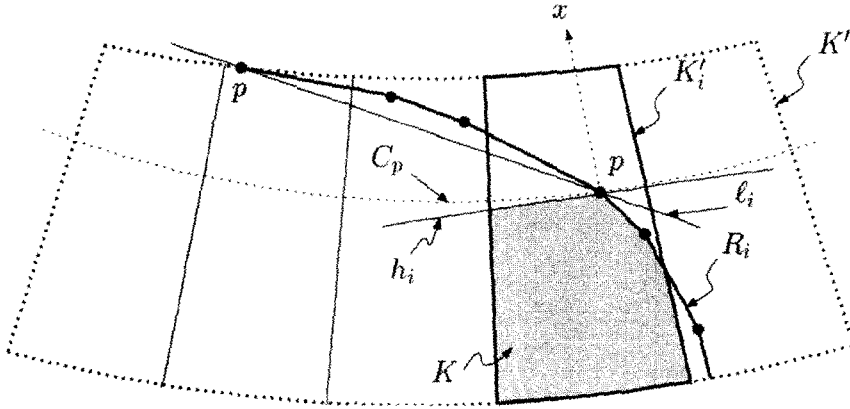


그림 4 사실 1의 증명

성립한다.

(증명) 트리 레벨에 대한 귀납법으로 증명한다. 우선 첫 번째와 두 번째 레벨에 대해선 알고리즘 구성방법에 따라 당연히 성립한다(그림 1(b)를 참조하라). 이제, 레벨 $j (> 2)$ 또는 그보다 작은 레벨에 대해선 위 사실이 성립한다고 가정하고, 다음 레벨 $(j+1)$ 의 영역 K 를 고려하자. 그림 4처럼 K 의 루트인 p 와 R -체인으로 연결된 레벨 j 의 영역 K' 의 루트를 p' 이라 하자. 영역 K 는 K' 이 m 개의 같은 각을 갖는 영역으로 분할된 영역들 중에서 i 번째 영역인 K'_i 의 부분영역이라 가정하자. 당연히 $K \subset K'_i \subset K'$ 이 성립한다. 정확한 설명을 위해, 몇 가지 기호를 정의하자. p 를 지나면서 직선 \overline{xp} 에 수직인 직선을 h_i 라 하고 x 를 포함하지 않는 반평면을 h_i^- 라 하자. p' 에서 시작하면서 p 를 지나고 있는 R -체인을 R_i 라 하자(만약 p 를 지나고 있는 R -체인이 두 개 이상이라면 가장 아래쪽 체인을 R_i 라 한다). R_i 의 아래쪽 영역을 R_i^- 라 하면, 알고리즘에 의해 $K = R_i^- \cap K'_i \cap h_i^-$ 으로 정의된다.

귀납법 가정에 의해 $|\overline{xp'}| \leq |\overline{xp}|$ 이다. 그림 4를 참조하자. 가정에 따라, $|\overline{xp'}| \leq |\overline{xp}|$ 이므로 p' 은 C_p 내부(또는 경계)에 존재한다. 직선 $\overline{p'p}$ 의 아래쪽 반평면을 ℓ 라 하자. 그러면 R_i 가 위로 볼록한 체인이기 때문에 $K \subseteq \ell^- \cap K'_i \cap h_i^-$ 이다. 같은 이유로 $(\ell^- \cap h_i^-)$ 과 디스크 C_p 는 점 p 를 제외한 어떠한 점에서도 교차하지 않는다. 즉, $\ell^- \cap h_i^-$ 에 속하는 모든 점 q 에 대해, $|\overline{xq}| \leq |\overline{xq}|$ 이 성립한다. 그런데 $K \subset (\ell^- \cap h_i^-)$ 이므로 K 에 속하는 모든 점 q 에 대해서도 $|\overline{xq}| \leq |\overline{xq}|$ 이 성립한다. (끝)

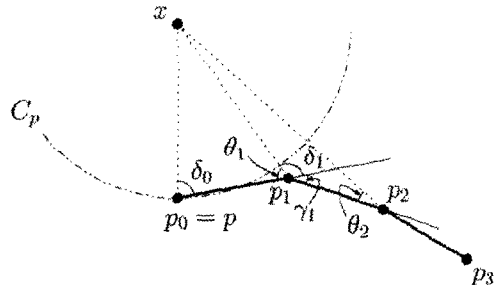


그림 5 사실 2의 증명

레벨 j 에 존재하는 루트가 p 인 영역 K 에 대해, 다음 레벨 $(j+1)$ 을 위해 $S(K)$ 에서 몇 개의 점들이 선택되고 p 와 함께 R -체인들로 연결된다. 이렇게 정의되는 R -체인 중 하나를(그림 5처럼) $R = \langle p_0 = p, p_1, \dots, p_t \rangle$ 이라 하자. R 은 p 에서 시작하여 p_1, p_2, \dots, p_t 점을 차례로 연결한 위로 볼록한 체인이다. 그러면 다음의 사실이 성립한다.

사실 2 레벨 j 에서 정의되는 R -체인 $R = \langle p_0 = p, p_1, \dots, p_t \rangle$ 을 고려하자. 그러면 모든 $0 \leq i < k < t$ 에 대해, $|\overline{xp_i}| \leq |\overline{xp_{i+1}}|$ 이 성립한다.

(증명) 이 사실은 임의의 $0 \leq i < t$ 에 대해, $|\overline{xp_i}| \leq |\overline{xp_{i+1}}|$ 이 성립함을 보이는 것과 같다. 우선 $|\overline{xp_0}| = |\overline{xp}| \leq |\overline{xp_1}|$ 임을 증명하자. 여기서 p_0 는 레벨 j 에 있는 영역 K 의 루트이고, p_1 은 K 에 포함되는 레벨 $j+1$ 의 어떤 영역의 루트이기 때문에 사실 1을 적용할 수 있다. 따라서 성립한다. 그림 5처럼 증명에 필요한 다양한 각도와 점을 정의한다. $|\overline{xp_0}| \leq |\overline{xp_1}|$ 이 이미 증명되었기 때문에, $\delta_0 \geq \theta_1$ 이고 $\delta_0 \leq \delta_1$ 이 된다. 따라서 $|\overline{xp_1}| \leq |\overline{xp_2}|$ 이 된다. 위로 볼록한 체인을 다루므로

항상 $\gamma_1 > 0$ 이다. 결국, $|\overline{xp_1}| \leq |\overline{xp_2}|$ 이 성립한다. 이 논리를 계속 적용할 수 있기 때문에 이웃한 두 점에 대해 $|\overline{xp_i}| \leq |\overline{xp_{i+1}}|$ 도 성립한다. (끝)

트리 T 의 루트 x 로부터 임의의 리프에 이르는 경로는 각 레벨에서 구성된 R -체인 또는 L -체인들로 구성되어 있으므로 위에서 증명한 사실 2를 이용하면 경로 위에 모든 점들이 루트 x 에 대해 단조함을 쉽게 증명할 수 있다. 이는 트리 T 의 단조하다는 의미이다.

소정리 1. 알고리즘에서 구성한 트리 T 는 루트 x 에 대해 단조하다.

분지수 분석: T 의 분지수를 분석해 보자. 현재 레벨 j 영역 K 의 루트 점 p 의 분지수를 살펴보자. 첫 번째로 p 와 같은 레벨 j 의 영역에서 선택된 점들이 서로 R -체인 또는 대칭적인 L -체인으로 연결되는데, 이때 체인이 p 에서 모두 시작한다면 분지수는 최대 $m-1$ 개로 증가하게 된다. 그 이유는 p 의 영역 K 를 제외한 $m-1$ 개의 영역들의 점들이 최대 $m-1$ 개의 R -체인 또는 L -체인에 의해 연결될 수 있고, 이 모든 체인이 p 에서 시작할 수 있기 때문이다. 두 번째는 다음 레벨 $(j+1)$ 에서 K 는 m 개의 부채꼴 영역으로 나뉘고 각 영역마다 한 점씩 해당 영역의 루트로 선택되어 p 의 자식 노드로 연결된다. 따라서 p 의 자식 노드를 위한 분지수는 최대 $m+1$ 을 넘지 않는다. 세 번째는 p 는 레벨 j 에서 정의되는 삼각주 영역의 루트 역할을 할 수도 있다. 그런데 p 가 놓인 레벨 $(j+1)$ 의 영역에서는 삼각주가 정의되지 않기 때문에, 분지수가 다시 최대 $m-1$ 만큼 증가할 수 있다. 마지막으로 p 가 이전 레벨 $(j-1)$ 의 (유일한) 부모와 연결해야 하므로 p 의 총 분지수는 $3m-1$ 을 넘지 않게 된다.

소정리 2. 알고리즘에서 구성한 트리 T 의 분지수는 임의의 $m \geq 2$ 에 대해 $3m-1$ 를 넘지 않는다.

지름 분석: 트리 T 의 루트 x 부터 레벨 $k (> 1)$ 에 있는 임의의 리프 노드 z 까지의 경로 $P_{x,z}$ 를 고려하자. 당연히 $P_{x,z}$ 는 유일하게 존재한다. 그러면 $P_{x,z}$ 위에 존재하는 레벨 j 의 영역 K_j 의 루트를 p_j 라 하자. (여기서 $p_1 = x$ 이고 $p_k = z$ 이다.) 전체 루트 x 에서 K_j 를 포함하는 최소 크기의 부채꼴의 각을 θ_j 라 하자(그림 6 참조). 주의할 것은 인접한 레벨 j 와 $(j+1)$ 의 두 점 p_j 와 p_{j+1} 이 반드시 $P_{x,z}$ 에서 인접할 필요는 없다는 것이다. 즉, 그림 5처럼, p_j 와 p_{j+1} 사이에 여러 점이 존재하여 R -체인 (또는 L -체인)으로 연결될 수도 있다. 이 R -체인을 P_j 라 하면, 당연히 $P_j \subseteq P_{x,z}$ 이다. 이제 루트 x 를

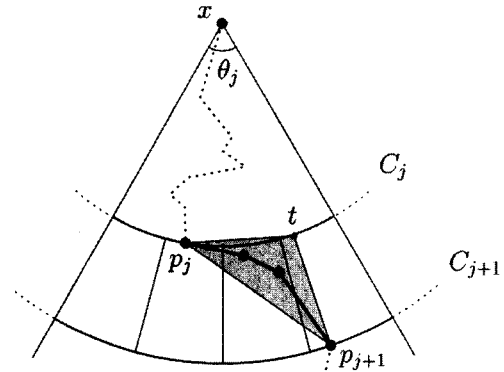


그림 6 소정리 3의 지름 분석을 위한 증명

중심으로 p_j 를 지나는 반지름이 d_j 인 디스크 C_j 와 p_{j+1} 를 지나는 반지름이 d_{j+1} 인 원 C_{j+1} 를 정의하고, 직선 $\overline{xp_{j+1}}$ 와 C_j 의 교차점이라 t 라 하자. 그러면 레벨 j 에서 레벨 $(j+1)$ 을 구성하는 방법에 의하면 R -체인 P_j 는 위로 볼록하고 루트 x 에 대해 단조하므로 P_j 는 삼각형 $\Delta(p_j, t, p_{j+1})$ 에 완전히 포함됨을 알 수 있다. P_j 의 위쪽으로 볼록한 성질과 삼각형에 관한 부등식 성질에 의해, $|P_j| \leq |\overline{p_{j+1}t}| + |\overline{tp_j}|$ 이 성립하고, $|\overline{tp_j}|$ 는 C_j 에서 각 θ_j 만큼의 원호의 길이보다 작기 때문에 $|\overline{tp_j}| \leq d_j \theta_j \leq D \theta_j$ 이 된다. 여기서 D 는 점 집합 S 의 지름임을 상기하자. 또한, $|\overline{p_{j+1}t}| = d_{j+1} - d_j$ 이다. 결국, 다음의 부등식이 성립한다.

$$\begin{aligned} |P_{x,z}| &\leq \sum_{j=1}^k |P_j| \\ &\leq \sum_{j=1}^k |\overline{p_{j+1}t}| + \sum_{j=1}^k |\overline{tp_j}| \\ &\leq \sum_{j=1}^k (d_{j+1} - d_j) + \sum_{j=1}^k d_j \theta_j \\ &\leq (d_k - d_1) + \sum_{j=1}^k D \theta_j \\ &\leq D(1 + \sum_{j=1}^k \theta_j). \end{aligned}$$

그런데 레벨 j 의 영역 K_j 는 다음 레벨에서 같은 각을 갖는 m 개의 영역으로 분할되므로 레벨 $(j+1)$ 에서의 정의되는 영역의 각 θ_{j+1} 은 θ_j/m 이 된다. 그러나 경우에 따라 삼각주 영역이 함께 고려될 수도 있기 때문에, 레벨 $(j+1)$ 에서 각 θ_{j+1} 은 실제로 $\theta_{j+1} \leq 2\theta_j/m$ 이 된다. 단, $\theta_1 = \pi/m$ 이다. 이 식을 반복해 정리하면 $\theta_j \leq \pi(2/m)^j$ 이 되고, 이를 위 부등식에 대입하면 아래와 같은 식이 성립한다.

$$\begin{aligned} |P_{x,z}| &\leq D(1 + \sum_{j=1}^k \theta_j) \\ &\leq D(1 + (\pi/2)((2/m)^1 + (2/m)^2 + \dots + (2/m)^\infty)) \\ &\leq D(1 + \pi/(m-2)) \end{aligned}$$

이제 T 의 지름 $D(T)$ 을 계산해보자. $D(T)$ 는 임의의 두 점 y, z 를 연결하는 T 의 경로 중에서 가장 긴 길이로 정의된다. 그러면 위에서 계산한 식과 S 에 대한 MDST의 지름 D^* 는 $D^* > D$ 라는 당연한 사실로부터 아래의 식이 성립한다.

$$D(T) = \max_{y,z} (|P_{x,z}| + |P_{x,y}|) \leq 2D(1 + \pi/(m-2)) \leq 2(1 + \pi/(m-2))D^*$$

결론은 T 의 지름은 최소 지름에 비해 $2(1 + \pi/(m-2))$ 배를 넘지 않는다는 것이다.

소정리 3. 알고리즘에서 구성한 트리 T 의 지름은 MDST 지름에 비해 $2(1 + \pi/(m-2))$ 배를 넘지 않는다.

위의 소정리 1부터 소정리 3을 정리하면 다음과 같은 본 논문의 주요 결과를 얻을 수 있다.

정리 1. 이차원 평면에 주어진 n 개의 점을 연결하는 신장 트리 중에서 분지수가 $3m-1$ 을 넘지 않고, 최소 지름에 비해 $2(1 + \pi/(m-2)) = O(1)$ 배를 넘지 않는 트리가 존재한다. 또한 이 트리는 루트에 대해 단조하며, 트리의 에지는 서로 교차하지 않는다. 여기서 $m \geq 2$ 인 자연수로 입력으로 주어진다.

위에서 설명한 방법은 점 집합 S 의 지름을 결정하는 두 점 중에서 한 점을 루트로 하여 트리를 구성하는 것이었다. S 의 임의의 점을 루트로 정하더라도 같은 방법을 적용하여 유사한 결과를 얻을 수 있다. 이 경우엔 루트가 된 점을 중심으로 그 점에서 가장 멀리 떨어진 점을 지나는 디스크를 C 로 정의하고, C 를 레벨마다 반복적으로 분할하면서 트리를 구성하게 된다. 이렇게 구성된 트리의 성질은 지름을 제외하고는 정리 1과 같다. 트리의 지름을 분석하는 방법도 유사하다. 유일한 차이점은 S 의 점들이 C 전체에 존재할 수 있다는 것이다. 이것은 정리 1의 지름 분석에서 π 대신 2π 로 대체하는 것과 같다. 그래서 다음의 정리가 성립한다.

정리 2. 이차원 평면에 주어진 n 개의 점 중에서 임의의 점을 루트로 하여 분지수가 $3m-1$ 을 넘지 않고, 최소 지름에 비해 $2(1 + 2\pi/(m-2)) = O(1)$ 배를 넘지 않는 트리가 존재한다. 또한 이 트리는 루트에 대해 단조하며, 트리의 에지는 서로 교차하지 않는다. 여기서 $m \geq 2$ 인 자연수로 입력으로 주어진다.

3. 결론

본 논문에서는 이차원 평면에 주어진 n 개의 점을 임의의 점을 루트로 하는 트리 구조 네트워크로 연결하였다. 이 트리는 분지수가 상수이고, 트리 지름이 MDST의 최소 지름에 비해 상수 배 이내이며, 루트 노드에 단조하고 에지는 서로 교차하지 않는 성질을 갖는다. 그러

나 분지수가 입력 매개 변수 $m \geq 2$ 에 대해, $3m-1$ 이므로 모든 가능한 분지수를 정확하게 조절할 순 없다. 이를 보다 자유롭게 조절할 수 있도록 트리를 구성하는 것이 향후 연구 과제 중 하나이며, 트리의 지름과 최소 지름의 비를 최대한 줄이는 것도 중요한 연구 과제이다. 마지막으로 이차원에서 수행된 연구 결과를 삼차원 이상의 고차원 문제로 확장하는 것도 매우 흥미로운 연구 주제이다.

참고 문헌

- [1] J.-M. Ho, D. T. Lee, C.-H. Chang, C. K. Wong, Minimum Diameter Spanning Trees and Related Problems, SIAM J. Comput. 20(5):987~997, 1991.
- [2] J. Gudmundsson, H. Haverkort, S.-M. Park, C.-S. Shin, and A. Wolff, Approximating the Geometric Minimum-Diameter Spanning Tree, APPROX 2002, Springer LNCS 2462, pp. 146~160, 2002.
- [3] S. Arya, G. Das, D. M. Mount, J. S. Salowe, and M. Smid, Euclidean Spanners: Short, Thin, and Lanky, Proc. Of STOC, pp. 26~37, 1995.
- [4] C. Monmna and S. Suri, Transitions in geometric spanning trees. Proc. 7th ACM Symp. Comp. Geom., 1991, pp. 239~249.



안희갑

1992년 3월~1996년 2월 경북대학교 컴퓨터 공학과 졸업(학사). 1996년 3월~1998년 2월 포항공과대학교 전자계산학과 졸업(석사). 1997년 9월~2000년 6월 Hong Kong Univ. of Science and Tech. Dept. Computer Science 수료(박사). 2000년 9월~2001년 12월 네덜란드 Utrecht Univ. Dept. Computer Science 졸업. 현재 2001년 10월~2004년 2월 한국과학기술연구원(KIST) 영상미디어센터 연구원. 2004.2~현재 한국과학기술원(KAIST)전산학과 BK21 연구교수. 관심분야는 Computational Geometry, Approximation Algorithms, Computer Graphics, Virtual Reality



신찬수

1991년 서울대 계산통계학과 학사. 1993년 한국과학기술원 전산학 석사. 1998년 한국과학기술원 전산학 박사. 1999년~2000년 홍콩과기대 전산학과연구원. 2000년~2001년 한국과학기술원 전산학과 연구교수. 2001년~현재 한국외국어대학 전자정보학부 디지털 정보공학 조교수. 관심분야는 계산기하학, GIS 알고리즘, 컴퓨터 그래픽스, 그래프 드로잉