

Robust Construction of Voronoi Diagram of Circles by Region-Expansion Algorithm

Donguk Kim[†]

Department of Industrial and Management Engineering, Gangneung-Wonju National University

영역 확장법을 통한 평면에서 원들의 보로노이 다이어그램의 강건한 계산

김 동 옥[†]

강릉원주대학교 산업경영공학과

This paper presents a numerically robust algorithm to construct a Voronoi diagram of circles in the plane. The circles are allowed to have intersections among them, but one circle cannot fully contain another circle. The Voronoi diagram is a tessellation of the plane into Voronoi regions of given circles. Each circle has its Voronoi region which is defined by a set of points in the plane closer to the circle than any other circles. The distance from a point p to a circle c_i of center p_i and radius r_i is $\|p-p_i\|-r_i$, which is the closest Euclidean distance from p to the circle boundary. The proposed algorithm first constructs the point Voronoi diagram of centers of given circles, then it enlarges each point to the circle and expands its Voronoi region accordingly. This region-expansion process is done by local modifications and after completing this process for the whole circles the desired circle Voronoi diagram can be obtained. The proposed algorithm is numerically robust and we provide with a few examples to show its robustness. The algorithm runs in $O(n^2)$ time in the worst case and $O(n)$ time on average where n is the number of the circles. The experiment shows that the region-expansion algorithm is robust and runs fast with strong linear time behavior.

Keywords : Voronoi Diagram of Circles, Robust Algorithm, Quasi-Triangulation, Additively Weighted Voronoi Diagram

1. 서 론

이차원 또는 삼차원 공간에 원형 또는 구형의 기하 요소가 있을 때, 이들의 보로노이 다이어그램이 구성되어 있다면 요소들 사이의 기하학적인 정보를 매우 효율적이면서 정확하게 파악할 수 있다. 예를 들어 특정 위치에서 가장 가까운 요소를 찾는 문제, 요소들 사이의 교차 여부를 파악하는 문제, 특정 요소가 가장 큰 영향을 미치는 영역을 정확하게 계산하는 문제, 특정 반지름의 물체가 원

형 장애물의 방해 없이 이동할 수 있는 경로 탐색 문제 등을 들 수 있다. 이와 함께 보로노이 다이어그램은 입자와 같이 원형으로 표현되거나 원형으로 근사화 할 수 있는 물체 또는 요소들에 대해서 위치 및 거리가 중요한 역할을 하는 문제에서 매우 다양하게 활용되어 왔다. 분자생물학 분야에서 단백질의 기하학적 구조를 분석하여 단백질의 특성 및 기능을 설명하는 것에 응용한 것은 대표적인 예가 된다[1, 4, 9]. 또한 평면에서 원형의 디스크 집합이 있을 때, 이들의 위치를 겹치지 않도록 이동하여 이들을 둘러싸는 최소 크기의 원을 계산하는 디스크 패킹 문제를 푸는데도 중요한 역할을 한다[14]. 원형 장애물이 있을 때 평면에서 임의의 두 점 사이의 장애물을 회피하는 최단 거리를 계산하는 문제 또한 응용 예로 들 수 있다[8].

Received 5 August 2019; Finally Revised 5 September 2019;
Accepted 6 September 2019

[†] Corresponding Author : donguk@gwnu.ac.kr

지금까지 원 집합 보로노이 다이어그램의 계산에 대한 많은 연구가 진행되어 왔으나[2, 3, 6, 7, 10, 12, 15], 다양한 응용문제에서 보로노이 다이어그램의 활용도를 높이기 위해서는 수치적 강건성이 보장되어야 한다. 보로노이 다이어그램의 안정적인 계산에 대해서는 크게 두 가지 접근법이 있다. 하나는 수치 계산 자체를 오차 없이 정확하게 하여 올바른 결과를 만들자는 것이고, 또 다른 접근법은 계산 과정 중 발생하는 수치 오차는 불가피한 것으로 보고 수치오차가 나타나더라도 위상 정보의 일관성(consistency)을 유지하도록 강제하는 위상 우선(topology-oriented) 방법이다.

점 집합에 대한 위상 우선 접근법에 대한 대표적인 연구로 Sugihara and Iri[13]의 연구를 들 수 있다. Sugihara and Iri는 점 집합 보로노이 다이어그램을 구할 때 점들을 하나씩 추가하면서 보로노이 다이어그램을 수정하는 점진적(incremental) 방식을 사용했는데, 점의 추가로 인하여 수정 되는 주변의 위상 요소들이 항상 일관성을 유지하도록 하는 방법을 발표하였다. 또한 Lee et al.[11]은 Sugihara and Iri의 접근법을 원 집합 보로노이 다이어그램의 수치적 안정성을 확보하는 곳에 적용하였다. Lee et al. [11]의 방법 또한 점진적으로 원들을 하나씩 추가하여 수정하는 방식이며, 하나씩 추가할 때 마다 보로노이 다이어그램의 위상 정보가 일관되게 유지될 수 있도록 하였다.

본 논문에서는 평면에서 원들의 보로노이 다이어그램을 수치적으로 안정되게 계산하는 또 다른 알고리즘을 제안한다. 이미 수치적으로 안정성 및 효율성이 확보되어 있는 원의 중심점들에 대한 보로노이 다이어그램으로부터 시작하며, 원을 하나씩 방문하여 반지름을 증가시킨다. 이에 따라 해당 보로노이 영역이 확장하게 되는데, 이를 수치적 안정성을 확보하는 방식으로 기존의 정보를 수정하는 방법을 제시한다. 이러한 보로노이 영역의 확장을 모든 원들에 대해서 수정하면 결과적으로 원들의 보로노이 다이어그램을 구할 수 있음을 본 논문에서 보여준다.

본 논문의 구성은 다음과 같다. 제 2장에서는 제안하는 알고리즘의 주요 배경 이론을 설명 한다. 제 3장에서는 제안하는 알고리즘의 핵심이 되는 영역 확장법을 설명한다. 제 4장에서는 알고리즘을 구현하여 실행한 결과를 보여주며, 제 5장에서 결론을 맺는다.

2. 배경 이론

2.1 점 집합 보로노이 다이어그램

평면에 점 집합 $P = \{p_1, p_2, \dots, p_n\}$ 이 주어져 있을 때, 평면을 각각의 주어진 점에 가장 가까운 영역들로 분할한

것을 점들의 보로노이 다이어그램(point Voronoi diagram)이라 한다. 점 집합 P 에 대하여 보로노이 다이어그램이 정의되기 때문에 P 의 각 원소를 제너레이터(generator)라 부른다. 각 제너레이터 p_i 의 보로노이 영역 $VR(p_i)$ 는 $\{p \mid \|p - p_i\| \leq \|p - p_j\|, \text{ for all } j \neq i\}$ 와 같이 전체 평면에서 다른 제너레이터들보다 p_i 에 더욱 가까운 점들의 집합으로 정의된다. 이때 점들의 보로노이 다이어그램 $VD(P)$ 는 $\{VR(p_1), VR(p_2), \dots, VR(p_n)\}$ 으로 정의된다. 보로노이 모서리는 이웃한 두 보로노이 영역들 사이의 교집합으로 정의되며 직선의 형태를 가진다. 이웃한 세 보로노이 영역의 교집합으로 정의되는 보로노이 꼭지점은 세 점들과 같은 거리를 가지며, 따라서 세 점을 지나는 원의 중심이 된다. 점들의 보로노이 다이어그램에 대해서는 매우 많은 연구가 진행되어왔으며, 안정적으로 실행되는 프로그램 코드를 공공 영역에서 쉽게 구할 수 있다.

2.2 원 집합 보로노이 다이어그램

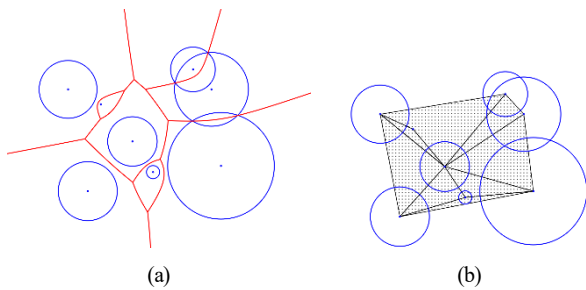
평면에 원 집합 $C = \{c_1, c_2, \dots, c_n\}$ 이 주어져 있을 때, 평면을 각각의 주어진 원에 가장 가까운 영역들로 분할한 것을 원들의 보로노이 다이어그램(circle Voronoi diagram)이라 한다. 원 집합 중 중심이 p_i 이고 반지름이 r_i 인 임의의 원 $c_i = (p_i, r_i)$ 에 대하여 c_i 의 보로노이 영역 $VR(c_i)$ 는 $\{p \mid \|p - p_i\| - r_i \leq \|p - p_j\| - r_j, \text{ for all } j \neq i\}$ 와 같이 정의된다. 이때 원의 보로노이 다이어그램 $VD(C)$ 는 보로노이 영역의 집합 $\{VR(c_1), VR(c_2), \dots, VR(c_n)\}$ 으로 볼 수 있다.

보로노이 모서리는 두 이웃한 보로노이 영역들 사이에서 보로노이 영역의 교집합으로 정의되는데, 두 원들 사이에 거리가 동일한 점들의 집합으로 볼 수 있다. $VD(C)$ 의 모서리는 쌍곡선(hyperbola)의 부분임이 잘 알려져 있다. 보로노이 꼭지점(vertex)은 셋 이상 이웃한 보로노이 영역의 교집합으로 해당 원들 사이에 거리가 같은 점으로 정의된다. 보로노이 꼭지점의 위치를 중심으로 하며 해당 원들에 동시에 접하는 원을 생각할 수 있는데, 이러한 원을 꼭지점 원(vertex circle)이라 하겠다.

본 논문에서는 설명의 편의를 위하여 각각의 보로노이 꼭지점은 세 개의 원들에 의해서 정의되는 것으로 가정한다. 네 개 이상의 원들이 하나의 공통 원에 접한다면 이에 대한 보로노이 꼭지점 하나가 공통 원의 중심에 정의될 수 있으며 이러한 상황을 디제너러시(degeneracy)라 부른다. 본 논문에서는 이러한 디제너러시 상황에서도 공통 원의 중심에 필요하다면 여러 개의 꼭지점과 길이가 0인 모서리를 채용하여 모든 꼭지점이 세 개의 모서리와 인접하도록 처리한다. 이를 통하여 사용자들이 디제너러시 상황을 위한 특별한 예외 처리가 없이 일반적인 상황으로 보로노이 다이어그램을 이용할 수 있도록 한다.

각각의 보로노이 꼭지점에 대하여 이를 정의하는 세 개 원들의 중심을 이어서 삼각형을 정의할 수 있고, 이들 삼각형들로 삼각화를 정의할 수 있다. 이러한 삼각화를 퀴지 삼각화(quasi-triangulation)라 부르며(이하 QT로 표현), $VD(C)$ 와 서로 쌍대(dual) 관계에 있다[5].

<Figure 1(a)>는 원들의 보로노이 다이어그램의 예를 보여준다. 붉은색의 모서리는 보로노이 모서리를 나타내며, 모서리들이 교차하는 점은 보로노이 꼭지점을 나타낸다. <Figure 1(b)>는 QT를 보여주는데, 보로노이 꼭지점 하나는 삼각형 하나와 대응되며, 보로노이 모서리는 삼각형의 모서리에 대응됨을 확인할 수 있다.

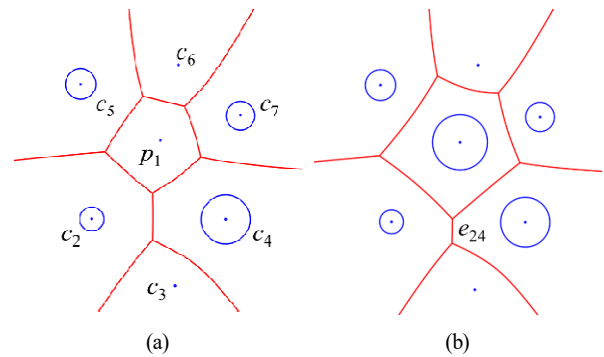


<Figure 1> Voronoi Diagram of Circles and Quasi-Triangulation

타나게 된다.

영역 확장법에서는 하나의 제너레이터가 확장할 때, 제너레이터의 반지름 범위 내에서 어떠한 종류의 이벤트가 어디에서 발생하는지를 알아내어서 이를 지역적으로 처리해주면 보로노이 다이어그램을 효율적이면서 위상의 통일성을 유지한 채로 수정할 수 있다는 것이 중심 아이디어이다.

본 논문에서는 확장 중인 보로노이 영역을 활성 보로노이 영역이라 하고 그때의 제너레이터를 활성 제너레이터라 하겠다. 모서리 e_{ab} 는 제너레이터 a 와 b 사이에 정의되는 모서리라 하겠다.



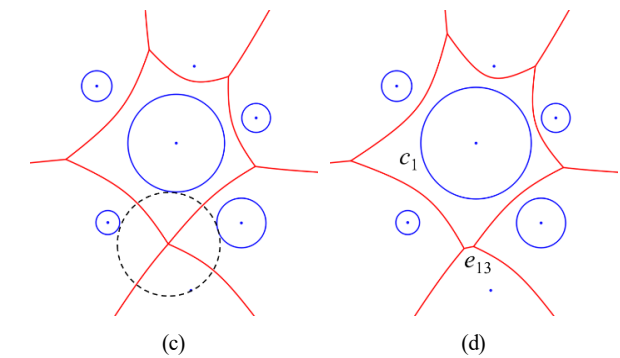
<Figure 2> Region-Expansion Process for Voronoi region of c_1 : (a) Initial Voronoi Diagram, (b) Enlarged Active Voronoi Region, (c) Active Generator Touches a Vertex Circle, (d) Voronoi Edge e_{24} is Flipped to e_{13}

3. 영역 확장법

본 절에서는 평면에서 원들의 보로노이 다이어그램의 계산을 위한 영역 확장법을 제안한다. 영역 확장법은 원들의 중심점들 P 의 보로노이 다이어그램을 계산한 다음 제너레이터 하나씩 반지름을 증가시켜 이에 따르는 제너레이터 주변의 변화만을 점진적으로 수정하는 방식이다. 이러한 과정을 모든 제너레이터에 대해 진행하여 최종적으로 원들의 보로노이 다이어그램을 계산하는 방법이다.

3.1 영역 확장 과정

<Figure 2>는 가운데 위치한 1번 제너레이터의 영역 확장 과정을 보여준다. 반지름을 확장하여 p_1 에서 c_1 이 될 때 보로노이 다이어그램이 변화하는 과정을 보여준다. 그림에서 볼 수 있듯이 하나의 제너레이터가 확장하면 그것의 보로노이 영역은 기존의 영역을 포함한 채로 넓어지게 된다. 처음에는 <Figure 2(b)>에서와 같이 보로노이 영역의 위상 변화 없이 영역만 넓어지다가 어느 순간부터는 보로노이 영역을 둘러싸는 모서리들에 대한 위상의 변화를 가져 오게 된다. 이러한 위상의 변화를 이벤트라고 한다. <Figure 2(c)>에서 기존의 모서리 e_{24} 가 사라지는 이벤트가 발생하고 그 이후 <Figure 2(d)>에서와 같이 e_{13} 가 새로 나



영역 확장 과정에서 활성 보로노이 영역을 둘러싸는 모서리들은 항상 그대로 유지가 된다. 또한 보로노이 영역이 단조 증가함에 따라 보로노이 영역의 경계에 꼭지점을 두면서도 경계를 구성하지 않는 모서리들은 길이가 보로노이 영역 쪽으로부터 점점 줄어들게 된다. 결국 현재의 활성 보로노이 영역에서 바로 다음으로 이벤트가 일어나게 된다면 이들 모서리들 중 하나의 길이가 0이 되면서임을 알 수 있다. 따라서 현재의 보로노이 영역의 경계로부터 바깥으로 뺀어 나가는 이러한 모서리들을 후보

모서리들이라 칭하겠고, 이벤트의 발생 여부는 이들 후보 모서리들의 변화를 통해서 판단할 수 있다.

<Figure 2(a)>에서 후보 모서리들은 e_{24} , e_{25} , e_{56} , e_{67} , e_{47} 이며, 영역 확장 중 e_{24} 의 길이가 0이 되면서 e_{24} 가 e_{13} 으로 플립(flip)되는 이벤트가 발생한다. 또한 <Figure 2(d)>에서와 같이 이벤트가 일어난 후에는 e_{24} 는 후보 모서리 집합에서 빠지게 되고 e_{23} 와 e_{34} 가 새로운 후보 모서리로 추가된다.

3.2 이벤트의 발생

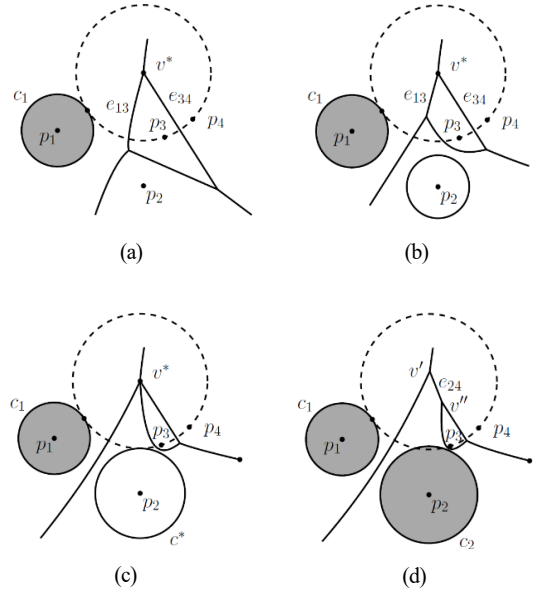
보로노이 다이어그램에서 보로노이 꼭지점의 주요 성질 중 하나는 다음과 같다. 보로노이 꼭지점은 가장 가까운 세 원과 동일한 거리에 위치하며, 보로노이 꼭지점을 중심으로 하고 세 원에 동시에 접하는 원을 꼭지점 원(vertex circle)이라 한다면 꼭지점 원의 내부는 다른 원과 교차하지 않는다는 것이다.

영역 확장 과정이 시작하는 시점에서는 이러한 무교차 특성이 모든 꼭지점에 대하여 성립되었지만 영역 확장이 진행되면서 해당 제너레이터가 기존의 꼭지점 원 중 하나와 교차하게 된다면 해당 꼭지점 원은 무교차 성질을 위반하게 되어 더 이상 존재할 수 없게 되며 위상 정보의 변화를 가져오게 된다. 이때가 이벤트가 발생하는 시점이 되며, 영역 확장 중 첫 번째 이벤트는 후보 모서리들의 꼭지점들 중 보로노이 영역의 경계를 이루지 않는 것들 중에서 일어난다. 이러한 꼭지점을 후보 꼭지점이라 칭하겠다.

후보 꼭지점의 꼭지점 원과 활성 제너레이터가 교차하게 되면 후보 모서리의 길이가 0이 되고 후보 모서리의 끝의 후보 꼭지점에서 무교차 특성을 위반하게 되어 위상의 변화가 일어나게 된다. 이때 기존의 후보 모서리는 활성 제너레이터의 보로노이 영역의 경계를 이루게 되며 이를 모서리 플립(edge flip)이라 칭한다.

<Figure 3>에서 v^* 는 후보 꼭지점을 나타내고 제너레이터 c_1 , p_3 , p_4 에 의해 정의된다. 점선의 원은 꼭지점 원을 나타내며 세 제너레이터에 접해 있음을 알 수 있다. 여기서 p_2 는 활성 제너레이터의 중심이며 c^* 는 활성 제너레이터가 꼭지점 원에 접할 때의 원이다. <Figure 3(a)>의 보로노이 다이어그램은 활성 제너레이터가 확장하기 전의 상태이며, <Figure 3(d)>는 제너레이터 p_2 의 확장이 끝나서 c_2 가 되었을 때의 보로노이 다이어그램을 보여준다.

후보 꼭지점에는 세 개의 인접 모서리가 있는데, 이들 중 후보 모서리는 하나 이상이 존재할 수 있다. 영역 확장 과정에서 모서리 플립을 하는 것은 하나만 존재하기 때문에 후보 모서리들 중 플립이 일어나는 것을 판별할 필요가 있다.



<Figure 3> Flip Condition for Candidate Edges Incident to a Candidate Vertex

예를 들어 <Figure 3(a)>에서 후보 꼭지점 v^* 에 인접한 세 모서리 중에 보로노이 영역 $VR(p_2)$ 에 인접한 후보 모서리는 e_{13} 와 e_{34} 이다. <Figure 3(d)>에서 점선으로 표시된 꼭지점 원이 c_2 와 교차하기 때문에 두 후보 모서리 중 하나는 플립이 일어난다. 그림에서는 e_{13} 가 보로노이 영역이 확장함에 따라 길이가 점점 짧아져서 <Figure 3(c)>에서 사라지게 되고 이후 <Figure 3(d)>에서와 같이 e_{24} 로 플립된다. 그러나 나머지 후보 모서리인 e_{34} 의 길이는 짧아지지만 플립 되지는 않는다.

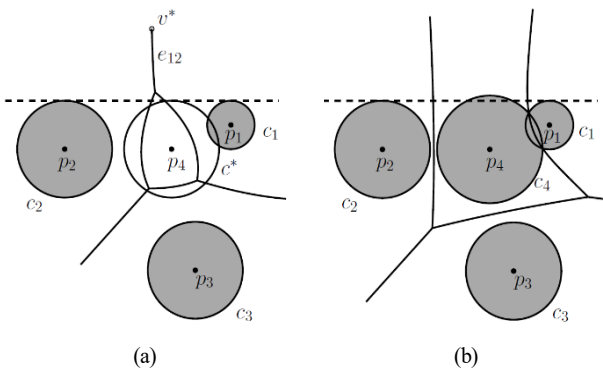
그러면 후보 모서리들 중 플립이 일어나는 모서리는 어떻게 판별할 수 있을까? 플립 모서리는 후보 꼭지점 원을 정의하는 세 개의 제너레이터와 함께 활성 제너레이터와의 상대적인 위치에 의해 결정된다. <Figure 3>에서와 같이 꼭지점 원은 이를 정의하는 세 개의 제너레이터에 동시에 접하는데, 꼭지점 원의 원주는 이 세 점들에 의해 세 개의 구역으로 구분할 수 있으며, 각 구역은 보로노이 모서리 하나와 대응된다. 예를 들어 <Figure 3>에서 c_1 의 점점과 p_3 사이의 구역은 모서리 e_{13} 와 대응하며, p_3 와 p_4 사이는 e_{34} 와 대응하며, p_4 와 c_1 의 점점 사이는 e_{14} 와 대응한다. 이때 만약 활성 제너레이터가 꼭지점 원을 접하게 된다면 세 구역 중 하나를 침범하게 될 것이며 해당 구역의 모서리 길이가 0이 되어 이후 플립된다. <Figure 3>에서 e_{13} 는 c_1 과 p_3 사이에서 정의되는데, <Figure 3(c)>에서와 같이 c^* 가 해당 구역을 터치하기 때문에 모서리 e_{13} 가 플립되며, 또다른 후보 모서리인 e_{34} 는 p_3 와 p_4 사이에서 그대로 존재하게 된다. 따라서 플립의 조건을 다음과 같이 정리할 수 있다.

모서리 e 가 플립하기 위해서는 다음의 두 조건을 모두 만족하여야 한다. (1) 후보 꼭지점의 꼭지점 원이 활성화 제너레이터와 교차한다. (2) 꼭지점 원과 활성화 제너레이터와의 접점이 e 를 정의하는 제너레이터들 사이의 구역에 위치한다.

이벤트의 발생 시점은 활성화 제너레이터가 꼭지점 원을 터치하는 순간이다. 이를 계산하는 것은 매우 간단하다. 후보 꼭지점에서 활성화 제너레이터까지의 거리에서 꼭지점 원의 반지름을 빼면 활성화 제너레이터가 터치하는 순간의 반지름을 쉽게 구할 수 있다. 만약 여기서 구한 반지름이 활성화 제너레이터의 반지름보다 작으면 이벤트가 일어남을 알 수 있다.

3.3 컨벡스헬 주변에서의 이벤트

만약 보로노이 모서리를 정의하는 두 개의 제너레이터가 원 집합에 대한 컨벡스헬(convex hull)의 경계를 이루고 있다면 해당 모서리는 바깥 방향으로 무한히 뻗어나가게 된다. 이때 무한히 뻗어나가는 방향에도 가상의 꼭지점이 존재하는 것으로 볼 수 있으며, 이러한 꼭지점을 무한 꼭지점이라 칭하겠다. 무한 꼭지점에서의 꼭지점 원의 경계에 해당하는 것은 두 원에 접하는 직선이다. 일반적으로 두 원에 접하는 직선은 네 개가 존재하나 무한 꼭지점에 해당하는 접선은 뻗어나가는 방향으로 원들 컨벡스헬의 경계를 일부 포함하는 직선이 됨을 쉽게 알 수 있다. 또한 꼭지점 원의 내부는 해당 접선을 경계로 하면서 제너레이터들을 포함하지 않는 반공간(half space)에 해당한다.



<Figure 4> Flip Condition for Candidate Edges Incident to an Infinite Vertex

<Figure 4>에는 이러한 예를 보여준다. 그림에서 p_4 는 활성화 제너레이터의 중심이며 네 개 제너레이터의 컨벡스헬 내부에 위치하기 때문에 닫힌(bounded) 보로노이 영역을 가지고 있다. 나머지 c_1, c_2, c_3 는 모두 컨벡스헬의 경계를 구성하고 있기 때문에 경계가 없는 무한(unbounded)의

영역을 가진다. 활성화 제너레이터가 컨벡스헬의 경계를 벗어나는 첫 번째 이벤트는 그림에서 점선으로 표현된 c_1 과 c_2 의 접선을 접할 때가 됨을 알 수 있다. 이 접선에 대응하는 보로노이 모서리와 꼭지점은 각각 e_{12} 와 v^* 이며, v^* 의 꼭지점 원은 점선의 접점을 경계로 하는 위쪽 반공간인데 <Figure 4(b)>에서와 같이 반공간을 침범하게 되면 v^* 는 더 이상 존재하지 못하고 e_{12} 는 플립 된다. 그 결과 확장 종료 후 c_4 또한 무한 영역을 가지며 컨벡스헬의 경계를 이루게 된다. 위에 설명한 내용을 다음의 모서리 플립 조건으로 정리할 수 있다.

무한 꼭지점을 한쪽 끝으로 가지는 모서리 e 가 플립하기 위해서는 다음의 두 조건을 모두 만족하여야 한다. (1) 후보 꼭지점의 반공간이 활성화 제너레이터와 교차한다. (2) 활성화 제너레이터와의 접점이 e 를 정의하는 제너레이터들 사이의 구역에 위치한다.

3.4 영역 확장 알고리즘

앞서 설명한 내용을 알고리즘으로 정리하면 다음과 같다. 제시하는 알고리즘은 하나의 제너레이터를 확장하는 것이고, 추후 이것을 모든 제너레이터들에 대해 적용하면 전체 원들의 보로노이 다이어그램을 구할 수 있다.

Algorithm 1: Region-Expansion(i, r_i)

Input: Voronoi diagram $VD(C_{i-1})$ where a circle set $C_{i-1} = \{c_1, c_2, \dots, c_{i-1}, p_i, p_{i+1}, \dots, p_n\}$

Procedure:

- 1 : Collect all candidate edges for the active generator p_i and store them into a queue Q.
- 2 : **while** Q is not empty **do** :
 - 2.1 : Pop an edge e from Q.
 - 2.2 : **if** e satisfies flip condition for radius r_i **do** :
 - 2.2.1 Flip e
 - 2.2.2 Push new candidate edge into Q.
- 3 : Update vertex circles and locations of Voronoi region boundary.

Output: current Voronoi diagram $VD(C_i)$

보로노이 다이어그램의 위상 정보가 Wigned-Edge 데이터 구조 등으로 저장되어 있다고 했을 때, Algorithm 1에서 단계 1은 후보 모서리의 개수가 m 이라고 한다면 $O(m)$ 의 시간이 소요된다. 단계 2.1은 $O(1)$ 의 시간이 필요하며, 단계 2.2.1에서 플립 연산 또한 $O(1)$ 의 시간이 소요되며, 2.2.2에서 새로운 두 개의 후보 모서리가 Q에

추가 된다. Q에 들어갈 수 있는 모서리의 최대 개수는 모든 나머지 제너레이터와 보로노이 모서리를 공유하는 개수인 $O(n)$ 이 된다. 따라서 단계 2는 최악의 경우 $O(n)$ 이 소요되며, 평균적으로 보로노이 영역이 가지는 평균 모서리 개수와 동일한 $O(1)$ 의 시간이 소요된다. 단계 3은 보로노이 영역의 경계를 구성하는 꼭지점의 개수는 $O(m)$ 이고 꼭지점 원을 계산하기 위해서는 $O(1)$ 의 시간이 소요된다. 후보 모서리의 개수 m 은 최악의 경우 $O(n)$ 이며 평균적으로 $O(1)$ 이다. 따라서 이들을 종합하면 Algorithm 1은 전체적으로 최악의 경우 $O(n)$ 시간이 소요된다.

Algorithm 2: Compute $VD(C)$ by Region-Expansion

Input: Voronoi diagram $VD(P)$ and a circle set $C = \{c_1, c_2, \dots, c_n\}$

Procedure:

- 1 : Insert all the generators into a queue Q.
- 2 : **while** Q is not empty **do** :
 - 2.1 : Pop a generator g .
 - 2.2 : $i \leftarrow$ index of g
 - 2.3 : Region-Expansion(i, r_i) to obtain $VD(C_i)$

Output: current Voronoi diagram $VD(C_n) = VD(C)$

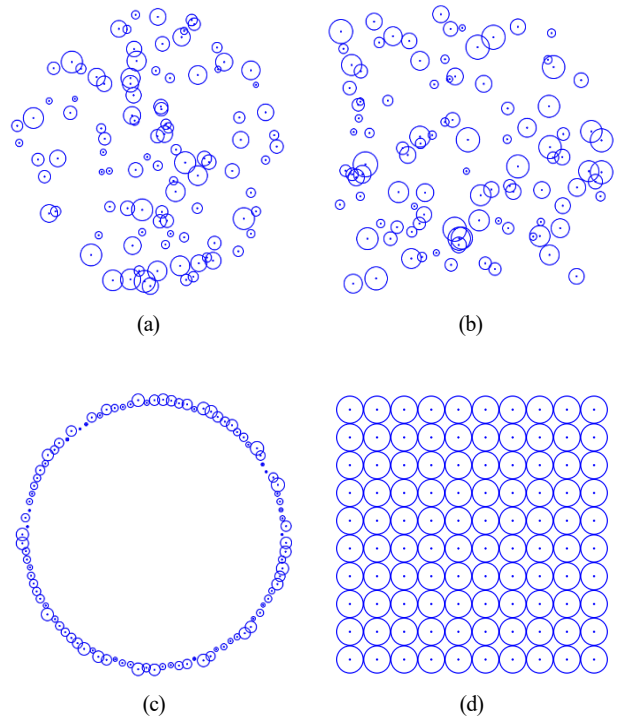
Algorithm 2는 Algorithm 1을 모든 제너레이터들에 대해 적용하는 것이기 때문에 전체적으로 최악의 경우 $O(n^2)$ 의 시간이 소요된다. 그러나 Algorithm 1이 평균적으로 $O(1)$ 시간을 필요로 하기 때문에 Algorithm 2는 제너레이터들이 전체적으로 균등하게 분포되어 있을 때 평균적으로 $O(n)$ 의 시간에 수행된다.

3.5 영역 확장 알고리즘의 수치적 안정성

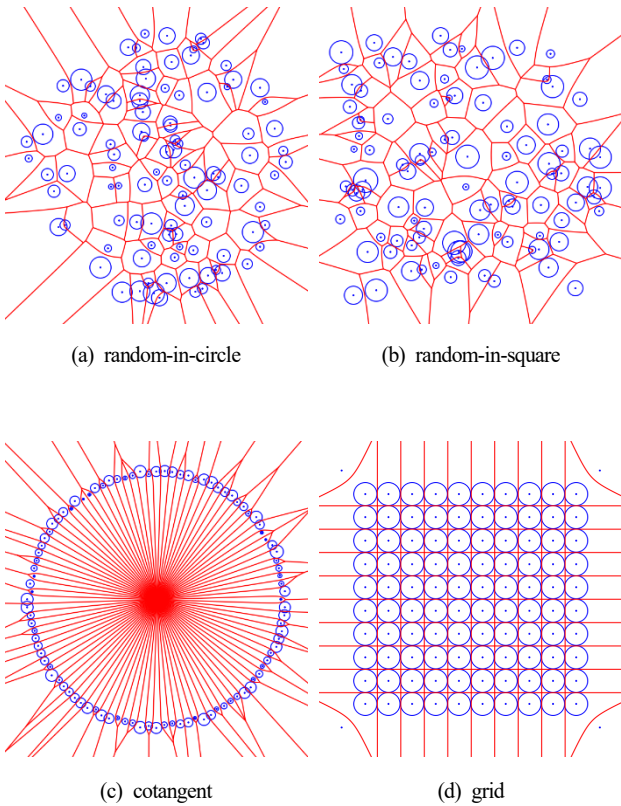
영역 확장 알고리즘은 일관성 있는 위상 구조를 이미 확보하고 있는 $VD(C_{i-1})$ 로부터 Algorithm 1인 영역 확장 과정을 통하여 $VD(C_i)$ 를 구한다. 영역 확장 과정 또한 현재의 위상 정보를 통하여 플립을 하여도 위상적 일관성이 유지가 가능한 모서리들만을 대상으로 플립이 필요하다면 플립을 시키기 때문에 항상 위상적 일관성을 유지한다. 따라서 만약 위상적 일관성이 확보된 점 집합 보로노이 다이어그램 $VD(P)$ 을 가지고 영역확장 과정을 진행한다면 수치적으로 매우 안정적으로 $VD(C)$ 를 구할 수 있다. 만약 플립 조건을 검사할 때 수치 오차로 인하여 잘못된 의사결정을 했을 지라도 플립 연산 자체는 위상적 연산이기 때문에 플립이 일어날 경우 위상적으로는 일관성을 유지할 수밖에 없다.

4. 구현 및 실험

제안하는 영역 확장 알고리즘을 C++로 구현하여 실험을 하였다. 실험 전체에서 한 원이 다른 원에 완전히 포함되지 않도록 생성 하였으며, 원들 사이의 교차는 허용하였다. 실험을 위하여 다음과 같은 4종류의 데이터 집합을 생성하였다. (1) 큰 원 내부에 분포한 무작위 원(random-in-circle), (2) 정사각형 내부에 분포한 무작위 원(random-in-square), (3) 하나의 원에 접하는 임의의 반지름의 원(cotangent), (4) 원의 중심이 격자에 위치한 원(grid). 데이터 집합 random-in-circle과 random-in-square의 원들은 모두 반지름 범위 $[0.2, 0.8]$ 에 무작위 분포를 하며, 데이터 집합 각각에 대하여 원들 면적의 합은 평균적으로 이들을 둘러싸는 컨테이너(원 또는 정사각형) 면적의 0.3이 되도록 설정하였다. 데이터 집합 cotangent는 디제너러시 정도가 매우 큰 경우이며, 집합 grid 또한 가까운 네 원의 중심에서 디제너러시가 발생하는 경우이다. 그리드의 외곽에 네 개의 점을 추가한 이유는 중심점에 대한 보로노이 다이어그램을 구할 때 수치적 안정성을 확보하기 위함이다. <Figure 5>에는 위의 4개 데이터 집합 각각에 대하여 100개 제너레이터로 구성된 예를 보여주고 있다.

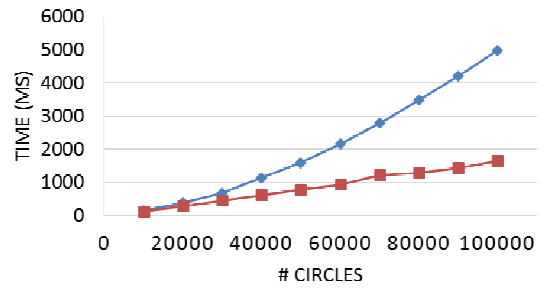


<Figure 5> Data Sets for Experiment : (a) 100 Random Circles in a Circular Container, (b) 100 Random Circles in a Square Container, (c) 100 Random Radii Circles Tangent to One Big Circle, and (d) 100 Circles on a Grid with 4 Bounding Points

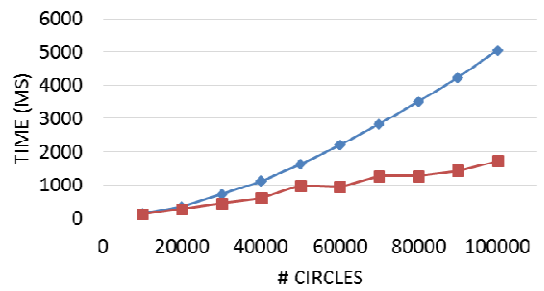


<Figure 6> Voronoi Diagram of Circles Constructed by the Region-Expansion Algorithm : (a) 100 Random Circles in a Circular Container, (b) 100 Random Circles in a Square Container, (c) 100 Random Radii Circles Tangent to One Big Circle, and (d) 100 Circles on a Grid with 4 Points

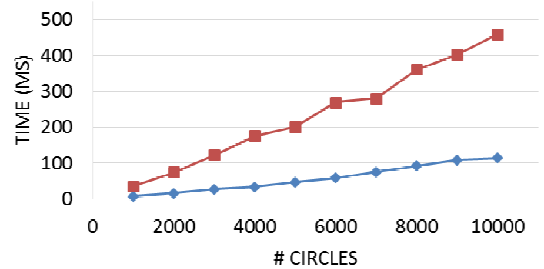
<Figure 6>은 제안하는 알고리즘으로 계산된 보로노이 다이어그램을 보여주고 있다. 네 개의 데이터 집합 모두에서 잘 계산된 것을 확인할 수 있다. 특히 세 번째 데이터 집합 cotangent의 경우 모든 원들이 하나의 원에 접하기 때문에 공통 원의 중심인 가운데 부분에서 위상적 일관성을 유지하기가 매우 어려움에도 불구하고 제안하는 알고리즘을 통하여 매우 안정적으로 계산이 된 것을 확인할 수 있다. 이론적으로는 가운데 꼭지점 하나에 보로노이 모서리 100개가 연결 되는 상황인데, 제안하는 알고리즘에서는 중심 부근에 많은 수의 꼭지점이 존재하고 각 꼭지점은 모두 3개의 모서리만을 인접하도록 구성을 한다. 데이터 집합 grid 또한 디제너러시에도 불구하고 계산이 잘 된 것을 볼 수 있고, 가까운 네 원 사이에는 보로노이 꼭지점이 하나에 네 개의 모서리가 인접한 것으로 보이나 그곳에는 위치가 같은 두 개의 꼭지점과 길이가 0인 모서리 하나가 존재하고 있어서 모든 꼭지점에서 3개의 모서리가 인접하고 있다.



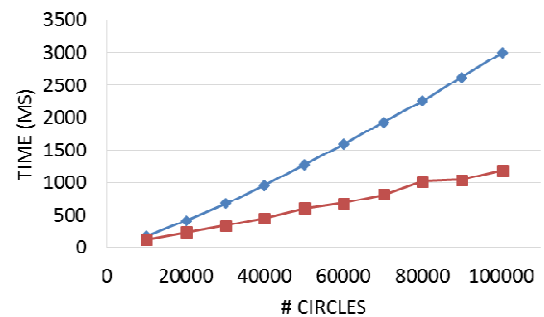
(a) random-in-circle



(b) random-in-square



(c) cotangent



(d) grid

<Figure 7> Computation Time of the Region-Expansion Algorithm (CPU : Intel Core2 Quad 2.83GHz, RAM : 3GB, OS : Windows 7 32 bit)

<Figure 7>은 실제 계산 시간을 보여준다. 알고리즘은 C++로 구현하였고, 실험 환경은 Intel Core2 Quad CPU, 3GB RAM, Windows 7 32bit OS이다. 그림에서 $VD(P)$ 는 원의 중심의 보로노이 다이어그램을 계산 하는데 걸린 시간이며, Region-Expansion은 $VD(P)$ 를 구한 이후 영역 확장 과정(Algorithm 2)을 수행할 때 소요된 시간이다. 따라서 전체 $VD(C)$ 를 계산하는 시간은 이들 두 시간을 합한 것이 된다. 실험에서 $VD(P)$ 를 구할 때에는 점진적(incremental) 알고리즘을 구현하여 이용하였고 강건화 방법론은 사용하지 않았다.

데이터 집합 모두에서 $VD(P)$ 이후의 영역 확장과정은 대체적으로 선형의 수행도를 보이고 있으며 이는 각 제너레이터의 영역 확장이 평균 $O(1)$ 의 시간에 이루어지고 있음을 설명한다. 또한 매우 빠르게 실행됨을 확인할 수 있다. Cotangent를 제외한 세 데이터 집합의 경우 $VD(P)$ 계산 시간을 포함하여 대략 1초에 30,000개의 원들을 처리할 수 있음을 확인할 수 있다.

제너레이터 하나당 평균 플립 횟수는 random-in-circle은 2.86, random-in-square는 2.88로 유사했으며, cotangent는 23.10, grid는 3.69로 나타났다. Cotangent의 경우는 꼭지점 원들이 크기와 위치가 모두 유사하기 때문에 영역 확장 과정에서 확장된 원이 많은 꼭지점 원과 교차하기 때문에 매우 많은 플립이 발생한다. 플립 횟수는 수행 시간과 매우 밀접한 관계를 가지는데 <Figure 7(c)>에서 보여지듯이 cotangent의 경우 다른 집합들과는 다르게 영역 확장 과정이 $VD(P)$ 보다 더욱 많은 계산 시간을 필요로 한다.

5. 결론

본 논문에서는 이차원 평면에서 유클리드 거리 기반의 원 집합 보로노이 다이어그램을 수치적으로 안정되게 구하기 위한 영역 확장법을 제안하였다. 본 논문을 통해서 영역 확장법은 이미 높은 수치적 안정성을 확보하고 있는 점 집합의 보로노이 다이어그램으로부터 제너레이터 하나씩 영역 확장을 할 때의 변화를 위상 연산을 통해서 수정하기 때문에 수치적으로 매우 안정적이면서 효율적으로 계산 할 수 있음을 설명하였고, 다양한 데이터 집합에 대한 실험을 통하여 수행 결과를 보여주었다. 실험 결과 영역 확장법은 매우 빠르고 안정적으로 수행되며, 입력 원의 개수에 대해서 선형의 수행도를 보여줌을 확인할 수 있었다. 본 논문에서 다루는 수치적 안정성을 확보한 원 집합 보로노이 다이어그램이 원형의 오브젝트와 관련된 다양한 응용문제를 푸는데 실질적인 기여를 할 수 있을 것으로 기대한다.

References

- [1] Cho, Y., Kim, J.-K., Ryu, J., Won, C.-I., Kim, C.-M., Kim, D., and Kim, D.-S., BetaMol : a molecular modeling, analysis and visualization software based on the beta-complex and the quasi-triangulation, *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, 2012, Vol. 6, No. 3, pp. 389-403.
- [2] Emiris, I.Z. and Karavelas, M.I.. The predicates of the Apollonius diagram : Algorithmic analysis and implementation, *Computational Geometry-Theory and Applications*, 2006, Vol. 33, pp. 18-57.
- [3] Fortune, S., A sweepline algorithm for Voronoi diagrams, *Algorithmica*, 1987, Vol. 2, pp. 153-174.
- [4] Kim, D., Cho, C.-H., Cho, Y., Ryu, J., Bhak, J., and Kim, D.-S., Pocket extraction on proteins via the Voronoi diagram of spheres, *Journal of Molecular Graphics & Modelling*, 2008, Vol. 26, No. 7, pp. 1104-1112.
- [5] Kim, D.-S. Kim, D., Cho, Y., and Sugihara, K., Quasi-triangulation and interworld data structure in three dimensions, *Computer-Aided Design*, 2006, Vol. 38, No. 7, pp. 808-819.
- [6] Kim, D.-S., Kim, D., and Sugihara, K., Voronoi diagram of a circle set from Voronoi diagram of a point set : I. topology, *Computer Aided Geometric Design*, 2001, Vol. 18, pp. 541-562.
- [7] Kim, D.-S., Kim, D., and Sugihara, K., Voronoi diagram of a circle set from Voronoi diagram of a point set : II. geometry, *Computer Aided Geometric Design*, 2001, Vol. 18, pp. 563-585.
- [8] Kim, D.-S., Yu, K., Cho, Y., Kim, D., and Yap, C., Shortest paths for disc obstacles, *Proceedings of the International Conference on Computational Science and Its Applications-ICCSA2004, Lecture Notes in Computer Science*, 2004, Vol. 3045, pp. 62-70.
- [9] Kim, J.-K. Cho, Y., Laskowski, R.A., Ryu, S.E. Sugihara, K., and Kim, D.-S., BetaVoid : molecular voids via beta-complexes and Voronoi diagrams, *Proteins : Structure, Functions, and Bioinformatics*, 2014, Vol. 82, No. 9, pp. 1829-1849.
- [10] Lee, D. and Drysdale, R., Generalization of Voronoi diagrams in the plane, *SIAM Journal on Computing*, 1981, Vol. 10, No. 1, pp. 73-87.
- [11] Lee, M., Sugihara, K., and Kim, D.-S., Topology-oriented incremental algorithm for the robust construction of the Voronoi diagrams of disks, *ACM Transactions on Mathe-*

- matical Software*, 2016, Vol. 43, No. 2, pp. 14:1-14:23.
- [12] Sharir, M., Intersection and closest-pair problems for a set of planar discs, *SLAM Journal on Computing*, 1985, Vol. 14, No. 2, pp. 448-468.
- [13] Sugihara, K. and Iri, M., A robust topology-oriented incremental algorithm for Voronoi diagrams, *International Journal of Computational Geometry & Applications*, 1994, Vol. 4, No. 2, pp. 179-228.
- [14] Sugihara, K., Sawai, M., Sano, H., Kim, D.-S., and Kim, D., Disk packing for the estimation of the size of a wire bundle, *Japan Journal of Industrial and Applied Mathematics*, 2004, Vol. 21, No. 3, pp. 259-278.
- [15] Yap, C.-K., An $O(n \log n)$ algorithm for the Voronoi diagram of a set of simple curve segments, *Discrete & Computational Geometry*, 1987, Vol. 2, pp. 365-393.

ORCIDDonguk Kim | <http://orcid.org/0000-0002-0510-922X>