

Triangulation of Voronoi Faces of Sphere Voronoi Diagram using Delaunay Refinement Algorithm

Donguk Kim[†]

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

딜러니 개선 알고리즘을 이용한 삼차원 구의 보로노이 곡면 삼각화

김 동 옥[†]

강릉원주대학교 산업경영공학과

Triangulation is one of the fundamental problems in computational geometry and computer graphics community, and it has huge application areas such as 3D printing, computer-aided engineering, surface reconstruction, surface visualization, and so on. The Delaunay refinement algorithm is a well-known method to generate quality triangular meshes when point cloud and/or constrained edges are given in two- or three-dimensional space. In this paper, we propose a simple but efficient algorithm to triangulate Voronoi surfaces of Voronoi diagram of spheres in 3-dimensional Euclidean space. The proposed algorithm is based on the Ruppert's Delaunay refinement algorithm, and we modified the algorithm to be applied to the triangulation of Voronoi surfaces in two ways. First, a new method to deciding the location of a newly added vertex on the surface in 3-dimensional space is proposed. Second, a new efficient but effective way of estimating approximation error between Voronoi surface and triangulation. Because the proposed algorithm generates a triangular mesh for Voronoi surfaces with guaranteed quality, users can control the level of quality of the resulting triangulation that their application problems require. We have implemented and tested the proposed algorithm for random non-intersecting spheres, and the experimental result shows the proposed algorithm produces quality triangulations on Voronoi surfaces satisfying the quality criterion.

Keywords : Voronoi Diagram, Voronoi Face, Delaunay Refinement Algorithm, Triangulation

1. 서 론

이차원 공간에 점들이 임의의 위치에 놓여 있을 때 이들 점들을 빠짐없이 연결하여 모서리 이외에는 겹치지 않는 삼각형들의 집합을 구성하는 것을 삼각화(triangulation)라 한다. 삼각화는 여러 가지 형태가 존재하는데, 그 중 딜러니 삼각화(Delaunay triangulation)는 보로노이 다이어그램

(Voronoi diagram)의 쌍대(dual) 구조인데, 삼각화를 구성하는 삼각형들 사이의 최소 각을 최대화 하는 특성을 가지고 있어 품질 높은 삼각형 메시(mesh)를 생성한다. 이런 이유로 컴퓨터 그래픽스, 3D 프린팅, 역공학, CAE 해석 등 다양한 응용 분야에서 널리 이용되고 있다[3, 10, 11]. 그러나 딜러니 삼각화 역시 주어진 점들의 상대적인 위치에 따라서 품질이 낮은 삼각형이 생성될 수도 있다. 이러한 문제의 해결을 위하여 주어진 꼭지점들의 위치를 이동시키거나 꼭지점들을 추가하여 삼각화를 개선하거나 최적화하는 방법이 존재한다[14].

Received 12 November 2018; Finally Revised 7 December 2018;
Accepted 10 December 2018

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

딜러니 개선 알고리즘(Delaunay refinement algorithm)이란 딜러니 삼각화가 주어져 있을 때, 품질 낮은 삼각형의 외접원 중심에 새로운 점을 추가하여 해당 삼각형 주변의 삼각화를 개선하고, 이를 반복적으로 수행하여 전체 딜러니 삼각화를 개선하는 방법이다. Chew와 Ruppert의 연구를 시작으로 많은 연구가 진행되고 있으며[1, 2, 6, 12, 13, 14], 개념 및 알고리즘이 단순하고 품질 높은 삼각형 메시(quality triangular mesh)를 효율적으로 생성할 수 있기 때문에 여러 응용 분야에 이용되고 있다.

그 중 Ruppert의 알고리즘은 삼각형들의 최소 각이 주어진 값 이상이 될 때까지 딜러니 개선 과정을 반복한다. 이를 통해 전체적으로 정삼각형에 가까운 삼각형들로 이루어진 삼각화를 얻을 수 있게 된다. 본 논문에서 제안하는 보로노이 곡면의 삼각화 방법은 Ruppert의 알고리즘을 기반으로 하는데, 보로노이 곡면의 고유한 특성을 반영하도록 수정하여 3차원 곡면의 품질 높은 삼각화를 구성하는 방법을 제안하고 구현 결과를 보여준다.

삼차원 구의 보로노이 다이어그램은 구들이 삼차원 유클리드 공간에 주어져 있을 때, 공간을 각각의 주어진 구에 가장 가까운 영역으로 분할한 구조를 말한다. 모든 물질은 입자로 이루어져 있고, 원자나 분자는 구의 형태로 모델링할 수 있기 때문에 재료과학, 분자 생물학 등의 많은 문제들은 구와 매우 밀접한 관련이 있다고 할 수 있다. 일단 구의 보로노이 다이어그램이 구해져 있다면 빈 공간의 분석, 특정 구가 영향을 미치는 영역 등과 같은 구와 관련된 공간의 분석을 매우 효율적으로 수행할 수 있다. 유클리드 공간의 구 보로노이 다이어그램의 알고리즘, 응용문제 등과 관련해서는 다음의 문헌을 참고하기 바란다[5, 7, 8].

구의 보로노이 다이어그램에 대한 보로노이 곡면의 삼각화 방법론으로 Will의 연구가 있다[15]. Will은 각각의 보로노이 영역에 대하여 이를 구성하는 보로노이 모서리들을 생성자의 중심에 위치한 단위 구에 사영하여 영역을 분할한 후, 분할된 구면을 삼각화하고 보로노이 곡면에 사영하여 곡면을 삼각화 하였다. 이를 통해 각 보로노이 영역을 가시화하고 영역이 차지하는 부피의 분포를 분석하였다[15]. Kim 등은 단백질에서의 상호작용면을 정의하고 분석하기 위하여 단백질을 구성하는 구의 보로노이 다이어그램을 이용하였다[4]. 이때 Kim 등은 상호작용면을 보로노이 곡면의 부분 집합으로 정의하였고 이를 가시화하기 위하여 해당 보로노이 곡면을 삼각화하였다.

본 논문은 다음과 같이 구성되어 있다. 제 2장에서는 배경 이론이 되는 보로노이 다이어그램과 딜러니 개선 알고리즘에 대해 개략적인 설명을 하고, 제 3장에서는 삼차원 구의 보로노이 곡면의 삼각화 방법을 제안한다. 제 4장과 제 5장에서는 제안하는 알고리즘의 구현 방법 및 실험 결과를 설명하고, 제 6장에서는 결론을 제시한다.

2. 배경 이론

본 절에서는 본 논문에서 다루고 있는 삼차원 구의 보로노이 다이어그램과 보로노이 곡면의 삼각화를 위해 이용되는 딜러니 개선 알고리즘을 설명한다.

2.1 구의 보로노이 다이어그램

3차원 공간에서 유클리드 거리 기반의 구의 보로노이 다이어그램은 다음과 같이 정의된다. 보로노이 생성자(generator) s_i 는 중심이 $\mathbf{P}_i = (x_i, y_i, z_i)$ 이고 반지름이 r_i 인 구로 정의된다. 생성자 집합 $S = \{s_i, i = 1, \dots, n\}$ 으로 정의한다. 보로노이 생성자 s_i 의 보로노이 영역은 $V(s_i) = \{\mathbf{P} \mid \|\mathbf{P} - \mathbf{P}_i\| - r_i \leq \|\mathbf{P} - \mathbf{P}_j\| - r_j, \forall j \neq i\}$ 으로 정의된다. 단, $\|\cdot\|$ 는 두 점 사이의 유클리드 거리를 나타낸다. 삼차원 구의 보로노이 다이어그램은 $VD(S) = \{V(s_i), i = 1, \dots, n\}$ 으로 정의할 수 있다. 삼차원에서 구의 보로노이 다이어그램의 알고리즘, 성질 및 구현에 관해서는 다음의 문헌을 참고하기 바란다[5, 8].

보로노이 곡면은 보로노이 영역의 경계를 이루는데, 가장 가까운 두 구 사이의 거리가 동일한 점들의 집합으로 정의된다. 두 구 s_i 와 s_j 사이의 보로노이 곡면이 존재한다면, 보로노이 곡면은 $VF(s_i, s_j) = \{\mathbf{P} \mid \|\mathbf{P} - \mathbf{P}_i\| - r_i = \|\mathbf{P} - \mathbf{P}_j\| - r_j\}$ 으로 정의된다. 점의 보로노이 다이어그램 $VD(P)$ 에서 보로노이 영역의 경계는 평면이 되는 것과는 달리 구의 보로노이 다이어그램의 보로노이 영역의 경계는 곡면을 이루며, 2엽 쌍곡면(two-sheet hyperboloid)의 일부가 된다는 성질은 잘 알려져 있다[5, 9].

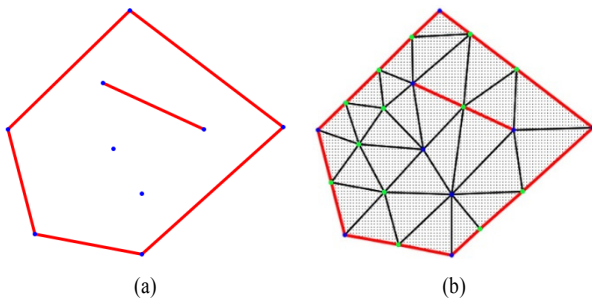
보로노이 곡선은 가장 가까운 세 구와의 거리가 동일한 점들의 집합으로 정의된다. 보로노이 곡선은 보로노이 곡면의 경계를 이루며, 기하학적 형태는 이를 정의하는 세 구의 상대적인 위치에 따라 직선, 쌍곡선, 포물선, 또는 타원의 형태가 된다. 이는 쌍곡면(보로노이 곡면)과 평면과의 교점으로도 정의할 수 있다.

2.2 딜러니 개선 알고리즘

딜러니 개선 알고리즘은 <Figure 1>에서와 같이 꼭지점 집합과 세그먼트(segment) 집합을 입력으로 할 때, 새로운 꼭지점을 세그먼트의 중심 또는 기존 삼각형의 외접원 중심에 추가하여 사용자가 요구하는 품질 조건을 만족하는 딜러니 삼각화를 구성하는 방법이다. 최종 삼각화의 꼭지점은 입력 꼭지점들을 모두 포함해야 하며, 삼각화의 모서리가 놓이는 영역은 입력된 세그먼트의 영역을 모두 포함해야 한다. 또한 만족시켜야 하는 품질 조건의 예로는 삼각화를 구성하는 모든 삼각형들 중 최대

모서리 길이, 최소 각도(minimum angle), 삼각형의 최대 크기 등이 가능하다.

<Figure 1>은 델러니 개선 알고리즘의 입력 요소들과 수행된 결과의 예를 보여준다. <Figure 1(a)>의 파란색 점은 입력 꼭지점을, 붉은색인 선분은 세그먼트를 나타낸다. <Figure 1(b)>는 알고리즘 수행 결과를 보여주는데, 최종 삼각화는 초기 꼭지점을 모두 포함하며, 초기 세그먼트는 여러 개의 하위 세그먼트로 분할될 수 있으며, 각 세그먼트는 삼각화를 구성하는 삼각형의 모서리를 이루어야 한다. 새로이 추가된 꼭지점들은 그림에서 녹색으로 표시하였고, 초기 입력된 꼭지점 이외에 세그먼트 위 9개와 나머지 3개의 꼭지점이 추가된 것을 알 수 있다. <Figure 1(b)>는 최소 각 30°를 품질 조건으로 적용한 결과이다. 그리하여 모든 삼각형들에 대하여 최소 각은 30° 이상이 된다.

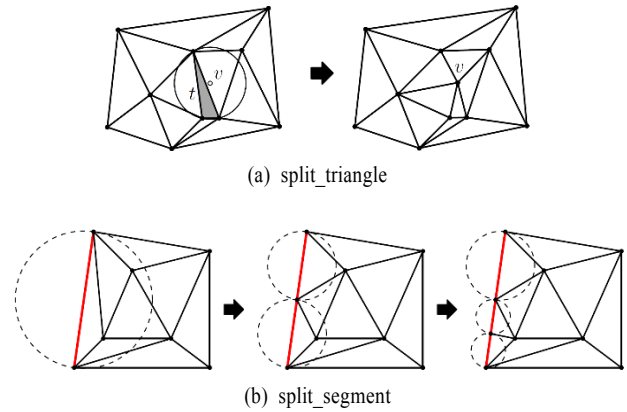


<Figure 1> Delaunay Refinement Algorithm : (a) Input Graph Consisting of 9 Vertices and 6 Segments, and (b) Resulting Triangulation with Minimum Angle of 30°

델러니 개선 알고리즘에서 두 가지 기본 연산은 split_triangle과 split_segment 연산이다. Split_triangle은 가장 핵심이 되는 연산으로, <Figure 2(a)>에서와 같이 얇은 삼각형 t 가 있을 때 삼각형의 외접원 중심 v 를 새로운 꼭지점으로 추가하고 지역적으로 수정하여 새로이 개선된 델러니 삼각화를 구성한다. 본 연산을 수행하면 삼각형 t 의 최소 각이 두 배가 되어 지역적으로 더 나은 삼각화를 얻을 수 있으며, 품질이 좋지 않은 삼각형들에 반복적으로 적용하면 전체적으로 높은 품질의 삼각화를 얻게 된다.

Split_segment 연산은 세그먼트 중심에 새로운 꼭지점을 추가하여 두 개의 세그먼트로 분할한다. 이후 지역적으로 수정하여 새로운 델러니 삼각화를 개선하는 것을 말한다. <Figure 2b>에서와 같이 세그먼트를 지름으로 하는 원을 세그먼트원(segment circle)이라 하고, 세그먼트원이 다른 꼭지점을 포함하면 해당 세그먼트는 위반한다고 표현한다. Split_segment 연산은 위반 세그먼트에 대하여 실행하며, 위반 세그먼트가 없어질 때까지 반복한다. 이러

한 과정을 <Figure 2(b)>에서 설명하고 있다. 첫 번째 그림에서 세그먼트원이 다른 두 꼭지점을 포함하기 때문에 split_segment 연산을 수행하여 가운데의 결과를 얻었고, 가운데 그림의 아래 세그먼트가 또다시 위반하기 때문에 split_segment 연산을 한 번 더 수행하여 우측의 결과를 얻었다.



<Figure 2> Two Primitive Operations of Delaunay Refinement Algorithm : (a) Split_Riangle, and (b) Split_Segment

델러니 개선 알고리즘을 요약하면 다음과 같다.

델러니 개선 알고리즘

입력 : 꼭지점 집합 V , 세그먼트 집합 S , 품질조건 q

- 델러니 삼각화 $DT(V)$ 계산
- 다음을 위반하는 세그먼트가 없고, 품질조건 q 를 만족할 때까지 반복
 1. 모든 위반 세그먼트에 대하여 반복적으로 split_segment
 2. 품질조건 q 를 만족하지 않는 삼각형 t 에 대하여 t 의 외접원의 중심점을 포함하는 모든 위반 세그먼트들에 대해 split_segment
 3. 위반 세그먼트가 없다면 split_triangle(t)

출력 : 현재의 삼각화

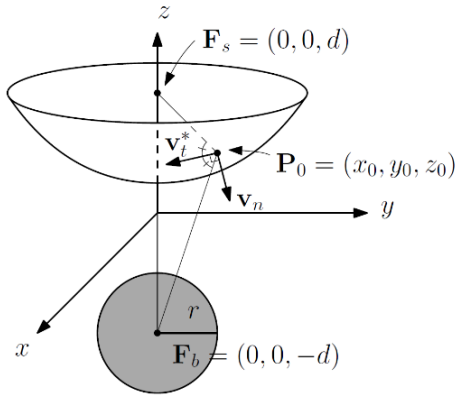
3. 방법

이번 절에서는 보로노이 곡면의 삼각화 방법을 제안한다.

3.1 보로노이 곡면의 삼각화

보로노이 곡면은 두 구와 동일한 거리를 가지는 점들 중에서 주어진 두 구를 가장 가까운 쌍으로 하는 점들의

집합으로 정의된다. 이는 두 구의 중심을 초점으로 하는 2엽 쌍곡면(two-sheet hyperboloid, 이후 쌍곡면으로 표현함) 중 반지름이 작은 구에 가까운 곡면의 일부가 되며 반지름이 큰 구의 방향으로 볼록하다. 두 구의 반지름을 동일한 크기만큼 줄여도 해당 쌍곡면은 그대로 유지가 되는데, <Figure 3>은 두 구의 반지름을 작은 구의 반지름 크기만큼 각각 줄였을 때의 쌍곡면을 보여준다. 만약 두 구의 크기가 동일하다면 두 구는 모두 점으로 줄어들 것이고 해당 곡면은 평면이 된다.



<Figure 3> Voronoi Face Transformed to the Canonical Space

3.1.1 표준 공간(Canonical Space)

각각의 보로노이 곡면은 삼차원 상에서 임의의 방향으로 존재하지만, 이를 표준화된 공간에 위치시키면 삼각화 계산이 매우 편리해진다.

표준 공간은 다음과 같이 정의된다. 대상이 되는 보로노이 곡면을 f 라 하자. 또한 f 를 정의하는 두 구를 $s_b = (P_b, r_b)$, $s_s = (P_s, r_s)$ 라 하자. 이때, P_b 와 P_s 는 구의 중심을 나타내고, r_b 와 r_s 는 반지름이며 $r_b \geq r_s \geq 0$ 이다. 두 구의 중심 P_b 와 P_s 의 중점을 P_m 이라 하자. 이때 표준 공간이란 (1) P_m 을 해당 공간의 원점으로 평행 이동 후, (2) 큰 구의 중심인 P_b 를 음의 z 축 상으로 작은 구의 중심을 양의 z 축 상으로 위치시키는 회전 변환을 통하여 이동한 공간으로 정의된다.

위의 변환을 통하면 $d = \|P_b - P_m\|$ 이라 할 때, 큰 구 s_b 의 중심은 표준 공간에서 $(0, 0, -d)$ 에 놓이게 되고, 작은 구 s_s 의 중심은 $(0, 0, d)$ 에 놓이게 된다. <Figure 3>은 표준 공간을 그림으로 설명하고 있다. 그림에서와 같이, 표준 공간에서는 반지름의 차이 $r = r_b - r_s$ 은 0보다 크거나 같으며, 만약 양수($r > 0$)인 경우 보로노이 곡면은 항상 $z > 0$ 공간에 존재하게 된다.

3.1.2 표준 공간에서 보로노이 곡면의 수식

표준 공간에서 보로노이 곡면 위의 점 $P = (x, y, z)$ 라

하고, 변환된 두 구의 중심을 $F_s = (0, 0, d)$ 와 $F_b = (0, 0, -d)$ 라 할 때, 다음의 수식이 성립한다.

$$\|P - F_s\| = \|P - F_b\| - r \tag{1}$$

식 (1)을 풀어 쓰면 다음과 같이 정리된다.

$$\sqrt{x^2 + y^2 + (z - d)^2} = \sqrt{x^2 + y^2 + (z + d)^2} - r \tag{2}$$

식 (2)에서 양 변의 거듭제곱을 통하여 근호를 제거하면 다음과 같은 음함수 형태의 수식으로 정리할 수 있다.

$$4r^2x^2 + 4r^2y^2 + (4r^2 - 16d^2)z^2 + 4r^2d^2 - r^4 = 0 \tag{3}$$

식 (3)은 표준 공간에서 xy -평면에 대칭인 2엽 쌍곡면이 되는데, 보로노이 곡면은 두 장의 곡면 중 $z > 0$ 의 공간에 놓이는 것이기 때문에 다음의 함수로 정리할 수 있다.

$$f(x, y) = \frac{1}{2} \sqrt{\frac{r^2(4x^2 + 4y^2 + 4d^2 - r^2)}{-r^2 + 4d^2}} \tag{4}$$

표준 공간에서 보로노이 곡면의 곡률(curvature)은 $x = y = 0$ 에서 다음과 같이 정의된다.

$$\kappa(r, d) = \frac{2r}{4d^2 - r^2} \tag{5}$$

3.1.3 표준 공간에서 보로노이 곡면의 삼각화

보로노이 곡면은 쌍곡면의 일부이기 때문에 한쪽으로만 볼록한 형태를 띠며, 표준 공간에서는 <Figure 3>에서와 같이 z 축의 음의 방향으로 볼록하다. 그렇기 때문에 보로노이 곡면 위의 점들이 주어질 때, 곡면의 삼각화는 삼차원 공간에서 곡면 위의 점들에 대한 볼록다면체(convex hull)를 구한 후 xy -평면에서 z 축 방향으로 보이는 경계 면들만을 취하면 얻을 수 있다. 또한 이는 표준 공간에서 보로노이 곡면 위의 점들을 xy -평면으로 사영한 후 딜러니 삼각화를 계산한 결과와 동일하다[11].

그렇기 때문에 제안하는 알고리즘은 표준 공간에서 보로노이 곡면 위의 점들을 xy -평면으로 사영한 후 딜러니 개선 알고리즘을 통하여 요구 품질을 만족하는 삼각화를 계산한 후, 이를 다시 식 (4)를 통하여 보로노이 곡면 위로 평행 이동하여 곡면의 삼각화를 얻는다.

3.1.4 평면에서의 삼각화와 차이점

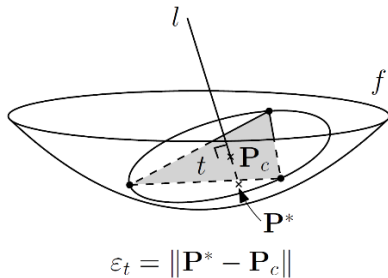
2차원 평면에서는 삼각화의 품질 척도로는 삼각형의 최소각, 삼각형의 면적, 삼각형의 개수 등 여러 가지가 있을 수 있다. 이 중 딜러니 개선 알고리즘에서 주로 삼각형의 최소각이 삼각화의 품질을 조절하는 파라미터의

역할을 한다.

본 논문에서 대상으로 하는 삼각화는 3차원 공간에서 곡면 위에서 이루어진다. 그렇기 때문에 2차원 평면에서와 마찬가지로 뾰족한 삼각형을 개선하여 더욱 정삼각형에 가까운 삼각형들로 구성되게 하는 것이 중요하다. 이와 더불어 본 논문에서는 보로노이 곡면을 근사화하는 것이기 때문에 대상 곡면과 결과 삼각화와의 오차가 품질을 결정하는 매우 중요한 요소가 된다.

3.2 삼각화 오차의 정의

보로노이 곡면을 삼각화하면 곡면과 삼각화 사이의 오차가 존재한다. 본 논문에서 제안하는 알고리즘은 곡면 오차의 허용치를 입력 값으로 받은 다음 이를 만족하는 삼각화를 생성하기 때문에 오차의 정의가 삼각화의 품질을 조절하는데 있어 매우 중요하다. 이러한 곡면의 오차는 다양한 방식으로 정의할 수 있다. 오차를 정의하는데 있어서 고려해야 하는 사항으로 두 가지를 들면 다음과 같다. (1) 실제 오차를 잘 반영해야 하며, (2) 계산이 쉬워야 한다. 이러한 사항을 고려하여 본 논문에서는 다음의 방식을 제안한다.



<Figure 4> Approximation Error of a Triangle t to a Voronoi Surface f

우선 삼각화를 구성하는 각각의 삼각형에 대하여 보로노이 곡면과의 오차를 정의할 수 있다. 삼각화를 구성하는 임의의 삼각형을 t 라고 할 때, t 의 세 꼭지점은 정의에 의해서 보로노이 곡면 위에 위치한다. <Figure 4>에서와 같이 삼각형의 외심을 P_c 라 할 때, P_c 를 통과하면서 t 에 수직인 직선을 l 이라 하자. 직선 l 과 곡면 f 의 교차점을 P^* 라 할 때, 삼각형 t 의 곡면 오차 ϵ_t 는 두 점 사이의 거리 $\|P^* - P_c\|$ 로 정의한다. 이때, P^* 는 삼각형 t 가 `split_triangle(t)` 연산에 의해 분할될 때 새로운 꼭지점의 위치로 이용된다. 그리고 P^* 를 xy -평면에 사영한 점이 $DT(V)$ 의 새로운 꼭지점이 된다.

제안하는 방법은 삼각형의 외심을 계산한 후 외심을 통과하는 직선과 곡면의 교점만을 계산하면 가능하므로

계산이 매우 간단한 장점이 있다. 또한 공간에서의 삼각화가 실제 곡면에 가까워질수록 삼각형과 교점 사이의 거리가 줄어들기 때문에 오차를 매우 잘 반영한다고 할 수 있다.

3.3 삼각화 알고리즘

보로노이 곡면의 삼각화 알고리즘은 곡면을 표준 공간으로 변환한 후, 경계모서리들을 분할하여 세그먼트를 구성하고 이들을 xy -평면으로 사영한다. 전처리 과정으로 사영된 세그먼트들을 입력으로 하는 초기 삼각화를 xy -평면에서 구성한 후, 반복적인 딜러니 개선 과정을 통하여 허용 오차를 만족시키는 삼각화를 구한다. 이를 요약하면 다음과 같다.

보로노이 곡면 삼각화

입력 : 보로노이 곡면 f , 경계 모서리 리스트 E , 허용오차 ϵ_b

[전처리]

1. 곡면 f 와 E 의 모든 모서리들을 표준 공간으로 변환
2. 변환된 E 의 모서리들을 표준 공간의 xy -평면으로 사영
3. 사영된 모서리들을 주어진 길이 이상의 선분으로 분할 후 세그먼트 리스트 S 에 추가
4. 딜러니 삼각화 $DT(V)$ 계산
5. 모든 위반 세그먼트에 대하여 반복적으로 `split_segment`

[반복]

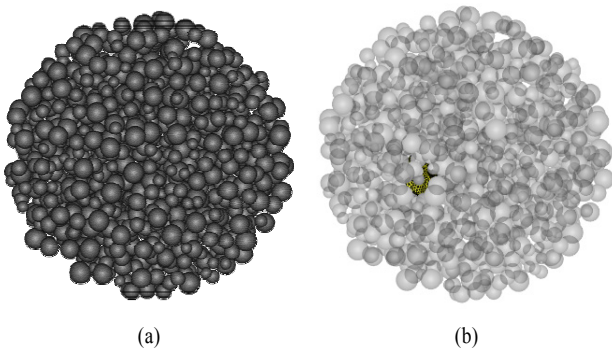
6. 위반 세그먼트가 없고, 모든 삼각형의 곡면 오차가 ϵ_b 보다 작아질 때까지 다음을 반복
 - 6.1 곡면오차가 ϵ_b 보다 큰 삼각형 t 에 대하여 새로운 꼭지점으로 인해 위반하는 모든 세그먼트들에 대해 `split_segment`
 - 6.2 위반 세그먼트가 없으면 `split_triangle(t)`

출력 : 현재의 $DT(V)$

경계 모서리를 통하여 생성된 세그먼트의 개수를 m 이라 할 때, 전처리과정은 5번 과정에서 최악의 경우 $O(m^2)$ 시간이 소요될 수 있다. 반면 6번 과정은 허용오차에 의해 반복 횟수가 결정되는데, 1회 반복 시 최대 $O(m)$ 시간이 소요될 수 있다. 그러나 참고문헌[6]에 소개된 버킷 구조를 활용한다면 전처리 과정은 평균적으로 $O(m)$ 시간에, 반복 과정은 1회당 평균 $O(1)$ 시간에 계산 가능하게 된다. 보로노이 곡면은 대부분의 경우 10개 미만의 경계 모서리를 가지고 있으므로 이론상 최악의 경우인 $O(m^2)$ 시간이 요구된다 하더라도 실제로는 매우 빠르게 계산된다.

4. 구현 및 실험

본 연구에서 제안하는 보로노이 곡면의 삼각화 알고리즘을 C++로 구현하여 실험하였다. 실험을 위하여 <Figure 5>에서와 같이 1,000개의 구를 컨테이너라 불리는 원점이 중심인 가상의 큰 구에 포함되도록 랜덤하게 생성하였다. 생성된 구들은 서로 교차가 없고 구들의 반지름은 범위[1, 2]에 확률적으로 균등하게 분포하도록 하였다. 컨테이너 구의 반지름은 대략 27.65이며, 생성된 구들의 밀도는 대략 0.178이다.



<Figure 5> The Sphere Set Used in the Experiment. 1,000 Non-Intersecting Random Spheres are Contained in a Big Spherical Container of Radius 27.65. Each Sphere's Radius is Uniformly Distributed in the Range of [1, 2], and the Packing Density in the container is 0.178 Approximately. (a) 1,000 Random Spheres in a Container, and (b) One Voronoi Region with the Spheres

보로노이 영역을 제안하는 방법으로 계산하기 위해서는 먼저 구의 보로노이 다이어그램을 계산한다. 실험에서는 모서리 추적 알고리즘[8]을 통하여 계산하고 모서

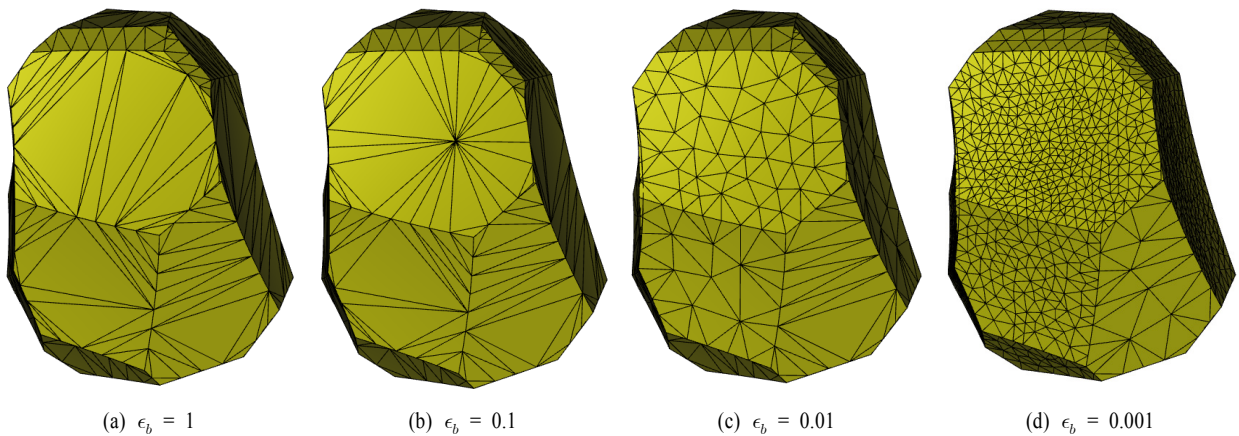
리의 수식을 계산하였다. 그런 다음 각각의 보로노이 곡면에 대하여 전처리 작업인 초기 삼각화를 계산하고 주어진 허용오차를 만족시키기 위하여 델러니 개선 과정을 반복한다.

5. 실험 결과 및 고찰

결과를 가시화하고 오차를 측정하기 위하여 <Figure 5(b)>에서와 같이 하나의 구를 임의로 선택하였고, 해당 구의 보로노이 영역을 구성하는 보로노이 곡면을 다른 구들과 함께 가시화 하였다. 그림의 보로노이 영역은 21개의 보로노이 곡면으로 구성되어 있고, 구들의 볼록다면체(convex hull) 내부에 위치하기 때문에 닫혀있다. 21개 곡면들의 곡률의 분포는 0과 1.11사이에 분포한다.

<Figure 6>에는 허용오차 ϵ_b 를 1, 0.1, 0.01, 0.001로 설정하고 이에 대한 삼각화 결과를 가시화 하였다. 그림에서 보여지듯이 허용오차가 작아질수록 더욱 많은 수의 삼각형으로 분할되며, 실제 쌍곡면에 가까워짐을 알 수 있다.

실험에서의 보로노이 영역은 서로 다른 곡률을 가지는데, 동일한 허용오차에 대해서도 곡률이 큰 곡면일수록 삼각형의 분포가 더욱 조밀해짐을 알 수 있다. 또 한 가지 주목할 점은 비록 제안하는 알고리즘이 생성된 삼각형의 최소 각을 입력값으로 받지 않음에도 불구하고 생성된 삼각형들은 곡률이 비교적 큰 곡면에 대하여 정삼각형에 가까워짐을 알 수 있다. 이는 뾰족한 형태의 나쁜 삼각형들은 비교적 길이가 길기 때문에 오차가 커지기 때문이며 델러니 개선 과정에서 새로 생기는 꼭지점이 삼각형의 무게 중심 부근에 위치하게 되어 새로 생기는 삼각형의 최소각이 기존의 것보다 커지게 되기 때문이다.



<Figure 6> Voronoi Region with Respect to the Error Bounds

<Table 1> Statistics of Voronoi Faces of a Voronoi Region after Refinement with Error Bound of 0.005. The Voronoi Region is shown in <Figure 6>

Face no.	Surface property				Initial triangulation					Triangulation after refinement				
	Radius difference	Distance between centers	Curvature	# Voronoi edges	Max. error	Min. angle (degree)	# triangles	# boundary edges	# vertices	Max. error	Min. angle (degree)	# triangles	# boundary edges	# vertices
1	0.83705	4.03244	0.10759	8	0.14314	6.26	32	34	34	0.00489	26.94	162	34	99
2	0.78559	4.01126	0.10154	5	0.08838	8.96	23	25	25	0.00486	22.62	98	26	63
3	0.67540	5.26807	0.04949	5	0.01195	16.42	14	16	16	0.00340	27.87	20	16	19
4	0.45915	4.45531	0.04676	7	0.06635	6.94	29	31	31	0.00489	18.98	93	31	63
5	0.44197	4.40282	0.04606	9	0.06547	6.80	33	35	35	0.00477	12.26	97	35	67
6	0.86300	6.28932	0.04447	4	0.01618	13.52	16	18	18	0.00435	28.97	24	18	22
7	0.44398	4.66889	0.04111	5	0.04652	6.79	25	27	27	0.00469	22.5	55	27	42
8	0.79353	6.81606	0.03463	3	0.00275	31.70	8	10	10	0.00275	31.7	8	10	10
9	0.60286	5.99343	0.03391	4	0.00530	22.95	12	14	14	0.00292	25.24	14	14	15
10	0.75875	6.86422	0.03261	5	0.00160	1.60	21	23	23	0.00160	1.6	21	23	23
11	0.52249	5.98095	0.02944	5	0.00897	12.67	17	19	19	0.00454	21.04	21	19	21
12	0.56922	6.26882	0.02921	5	0.00643	16.67	15	17	17	0.00389	23.64	17	17	18
13	0.32679	4.84887	0.02793	8	0.03460	7.49	29	31	31	0.00469	15.09	55	31	44
14	0.23518	4.94993	0.01924	8	0.03233	6.31	31	33	33	0.00455	7.65	51	33	43
15	0.40106	6.76709	0.01758	4	0.00136	29.94	14	16	16	0.00136	29.94	14	16	16
16	0.45485	7.58746	0.01586	3	0.00005	9.88	11	13	13	0.00005	9.88	11	13	13
17	0.34013	7.16215	0.01329	5	0.00396	0.03	15	17	17	0.00396	0.03	15	17	17
18	0.07355	6.27316	0.00374	7	0.00301	6.19	26	28	28	0.00301	6.19	26	28	28
19	0.07828	6.94184	0.00325	7	0.00408	8.07	27	29	29	0.00408	8.07	27	29	29
20	0.05738	7.69922	0.00194	3	0.00004	8.75	1	3	3	0.00004	8.75	1	3	3
21	0.03882	7.65222	0.00133	4	0.00026	18.64	15	17	17	0.00026	18.64	15	17	17

<Table 1>은 실험에서의 보로노이 영역을 구성하는 곡면들의 허용오차를 $\epsilon_b = 0.005$ 로 설정했을 때, 초기 삼각화와 제안하는 방법으로 새롭게 구성한 삼각화의 데이터를 보여준다. 총 21개 곡면에 대하여 곡률의 내림차순으로 정렬하였고, 곡률은 두 구의 반지름 차이와 중심 사이의 거리에 의해 결정되며 식 (5)를 이용하여 계산하였다.

<Table 1>에서 초기 삼각화의 최대 오차는 곡면 번호 1에서 0.14314이며, 이는 <Figure 6(a)>의 $\epsilon_b = 1$ 보다 작다. 그러므로 이 그림은 초기 삼각화와 일치하게 된다. <Figure 6(b)>에서 허용 오차는 $\epsilon_b = 0.1$ 인데, 이를 만족하지 못하는 곡면은 곡면 번호 1만 해당하며, 나머지 곡면은 딜러니 개선 과정을 진행할 필요가 없다. <Figure 6(b)>에서 정면 상단의 넓은 곡면이 곡면 번호 1에 해당하며 가장 곡률이 크다. 하나의 꼭지점이 딜러니 개선 과정을 통하여 새로이 추가되어 개선된 것을 그림으로 확인할 수 있다. 곡면 번호 15부터 21까지의 7개 곡면은 상대적으로 작은 곡률을 가지게 되어 초기삼각화의 최대 오차가 허용오차보다 작게 되어 추가적인 딜러니 개선 작업이 필요하지 않다. 그리고 곡면 번호 8과 10은 상대적으로 큰 곡률을 가지고 있으나 보로노이 곡면이 되는 부분이 중심(즉, 표준 공간에서 z축)에서 먼 곳에 위치해 있고 그 넓이가 작아 최대 오차가 허용오차보다 작다.

6. 결 론

본 논문에서는 삼차원 구의 보로노이 다이어그램의 보로노이 곡면의 삼각화 방법을 제안하였다. 제안하는 방법은 보로노이 곡면을 표준 공간으로 변환한 후 초기 딜러니 삼각화를 구성한 후 딜러니 개선 알고리즘을 허용오차를 만족할 때까지 반복한다. 실제 곡면과 삼각화된 곡면 사이의 오차를 근사화하는 정의하는 새로운 방법을 제시하여 적용하였다. 그 결과 주어진 허용오차를 만족하는 삼각화된 보로노이 곡면을 효과적으로 구할 수 있었다. 제시하는 방법론은 간단한 기초 연산인 split_segment과 split_triangle만을 이용하기 때문에 구현이 쉽고 수치적으로 안정적인 장점이 있다. 또한 제안하는 방법은 보로노이 곡면뿐만 아니라 일반적인 자유 곡면의 삼각화에도 적용 가능할 것으로 기대하며, 특히 하나의 평면으로 투영 가능한 2차 곡면에 대해서는 곧바로 적용 가능할 것으로 판단한다.

Acknowledgement

This research was supported by the Research Grant of Gangneung-Wonju National University.

References

- [1] Chew, L.P., Guaranteed-Quality Mesh Generation for Curved Surfaces, *Proceedings of the 9th Annual Symposium on Computational Geometry*, 1993, pp. 274-280.
- [2] Chew, L.P., Guaranteed-Quality Triangular Meshes, Technical Report TR-89-983, Cornell University, 1989.
- [3] de Berg, M., van Kreveld, M., Overmars, M., and Schwarzkopf, O., *Computational Geometry : Algorithms and Applications*. Springer, Berlin, 2nd edition, 2000.
- [4] Kim, C.-M., Won, C.-I., Cho, Y., Kim, D., Lee, S., Bhak, J., and Kim, D.-S., Interaction Interfaces in Proteins via the Voronoi Diagram of Atoms, *Computer-Aided Design*, 2006, Vol. 38, No. 11, pp. 1192-1204.
- [5] Kim, D. and Kim, D.-S., Region-Expansion for the Voronoi Diagram of 3D Spheres, *Computer-Aided Design*, 2006, Vol. 38, No. 5, pp. 417-430.
- [6] Kim, D., Acceleration of Delaunay Refinement Algorithm by Geometric Hashing, *Korean Journal of Computational Design and Engineering*, 2017, Vol. 22, No. 2, pp. 110 - 117.
- [7] Kim, D., Using Voronoi Diagram and Power Diagram in Application Problems, *Journal of Society of Korea Industrial and Systems Engineering*, 2012, Vol. 35, No. 4, pp. 235-243.
- [8] Kim, D.-S., Cho, Y., and Kim, D., Euclidean Voronoi Diagram of 3D Balls and Its Computation via Tracing Edges, *Computer-Aided Design*, 2005, Vol. 37, No. 13, pp. 1412-1424.
- [9] 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.
- [10] Lee, K., *Principles of CAD/CAM/CAE Systems*, Addison-Wesley, 1999.
- [11] Okabe, A., Boots, B., Sugihara, K., and Chiu, S.N., *Spatial Tessellations : Concepts and Applications of Voronoi Diagrams*, 2nd edition, Chichester, John Wiley & Sons, 2000.
- [12] Ruppert, J., A Delaunay Refinement Algorithm for Quality 2-Dimensional Mesh Generation, *Journal of Algorithms*, 1995, Vol. 18, No. 3, pp. 548-585.
- [13] Shewchuk, J.R., Delaunay Refinement Algorithms for Triangular Mesh Generation, *Computational Geometry-Theory and Applications*, 2002, Vol. 22, pp. 21-74.
- [14] Shewchuk, J.R., *Lecture Notes on Delaunay Mesh Generation*, Univ. of California at Berkeley, 2012.
- [15] Will, H.-M., *Computation of Additively Weighted Voronoi Cells for Applications in Molecular Biology*, Ph.D. dissertation, Swiss Federal Institute of Technology, Zurich, Switzerland, 1999.

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