□特 輯□

삼각분할(Triangulation)의 최근 연구 동향

이화여자대학교 전자계산학과 이상호* · 신금림**

목 차

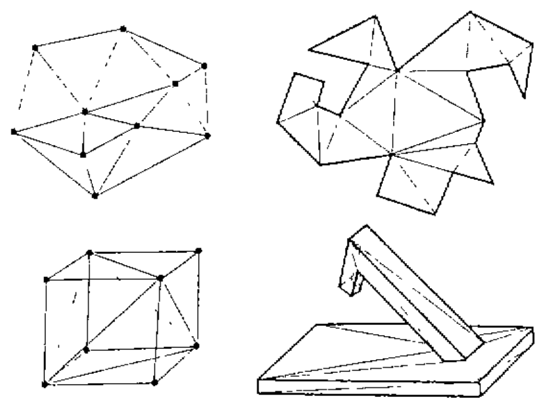
Ⅰ. 서 론

삼각분할(Triangulation)은 (그림 1)의 예에서 볼 수 있듯이 점집합, 다각형, 혹은 다면체 등의 분할 대상을 분할 후의 인접 영역이 정점이나 저차원의 면만을 공유하도록 삼각형 혹은 사면체로 나누는 것이다.

삼각분할 문제는 주로 분할할 대상의 특성에 따라 계산기하학의 여러 분야 중에서 근접(Proximity) 또는 분할(Decomposition) 문제의 일부로 다루어진다[70]. 근접 문제는 모든 점들의 근접성 정보를 찾아 처리하고 적절히 기억하여 전체 집합의 전반적인 특성이나 형태에 관련된 문제를 해결하려는 연구로 가장 가까운 이웃을 찾는 문제(Nearest neighbor problem), 점이 포함되는 영역을 찾는 문제(Point location problem) 등의 계산기하학 자체에 관한 문제 이외에 유한 원소 분석(Finite element analysis). 표면 모델 구성, 그래픽스 등의 모델링 기법에 이용된다. 반면, 분할 문제는 주로 다른 기하학적 문제를 위한 전처리 단계로 제공되어 복잡한 문제를 여러 단순한 문제의 모임으로 바꾸어 해결하고자 할 때 이용될 수 있으며 별형(Star-shaped). 볼록 다각형(Convex polygon), 단조(Monotone) 다각형으로의 분할 등이 있으나[6, 43,51,60,72], 이 중 삼각형이 구조적으로 저장 및 처리가 간단하여 많이 다루어져 왔다[14,43]. 이의 이용분야로는 가시성 및 최단경로 관련문제, 화랑문제 등 계산기하학의 여러 영역과 CAD, 컴퓨터 그래픽스, 패턴 인식 등이 있다.

일반적으로 어떤 대상의 분할방식으로 다음 세 가지를 생각해 볼 수 있다[2].

(1) 입력점 외의 가상점(Steiner point)을 허용하는 분할과 그렇지 않은 분할 : 가상점을 허용하는 경우는 추



(그림 1) 2, 3차원에서 삼각분할의 예

* 종신회원
** 준회원

가할 점의 수 및 위치의 결정 때문에 알고리즘이 복잡하고 $O(n^2)$까지의 원소들로 분할될 수 있으므로 패턴인식 등에서 많은 원소를 요구하지 않는 경우는 점의 추가를 허용하지 않는 것이 좋다. 반면, 가상점의 추가를 허용할 경우는 구성원소의 특성을 일관성 있게 분할할 수 있다는 장점을 가진다. 삼각분할의 경우, 점의 추가를 허용할 경우를 Steiner 삼각분할이라 하고 점의 추가를 허용하지 않을 경우를 chordal 삼각분할이라 한다.

(2) 분할된 구성요소들의 중복(Overlap)을 허용하는 분할과 허용하지 않는 분할 : 분할 후의 구성요소들이 겹치는 것을 허용하는 분할을 Covering이라 하고 d차원 분할에서 d보다 작은 차원의 공통면(d=3일 경우, 점, 선분, 면)에서만 중복을 허용하는 분할을 Partition이라 한다. Covering은 분할 후 구성요소의 모양을 일정하게 한다거나 구성요소의 수를 제한할 필요가 있을 경우 이용되며, 구성요소간 겹치는 부분의 기억 및 처리를 어떻게 할 것인가가 문제가 된다. Partition은 구성요소간의 경계가 명확하므로 구성요소별 처리가 쉬우나, 각 구성요소의 모양 및 크기 등이 일정하지 않다.

(3) 구성요소를 중심으로 한(Component-oriented) 분할과 절차를 중심으로 한(Procedure-oriented) 분할 : 구성요소 중심인 경우는 구성요소의 명확한 정의에 따라 분할하는 것이고, 절차 중심인 경우는 주어진 절차에 의한 분할로서 정확한 구성요소의 특성은 알 수 없다. 구성요소 중심인 경우는 위에서 언급한 일반적인 분할이 있고, 절차 중심의 방법으로는 Relative Neighborhood Graph(RNG)에 의한 분할[2,37]과, Gabriel Graph(GG)에 의한 분할[2] 등이 있다.

이 글에서는 가상점을 허용하지 않고 접침이 없는 삼각분할을 주요 대상으로 다루어 언급한다. 따라서 본문에서 분할은 특별한 언급이 없는 한 partition을 의미한다.

일반적으로 삼각분할의 대상이 되는 것은 점집합, 선분집합, 다각형, 다면체 등이 있는데, 어떤 것을 분할하느냐에 따라 삼각분할의 결과가 달라질 뿐 아니라, 그들이 어떤 차원에 놓이느냐에 따라서도 그 특성 및 접근법이 달라진다. 같은 대상에 대해 삼각분할 알고리즘을 선택할 때 고려되는 사항은 다음과 같이 크게 세 가지로 볼 수 있다[58].

- 삼각분할의 질(Quality)
- 삼각분할 알고리즘의 계산복잡도
- 알고리즘의 일반성

초기에 삼각분할 문제가 대두될 때에는 하나의 대상에 대해 보다 빠르고 효율적인 알고리즘을 구하려는 연구 위주였으나, 최근에는 이와 함께 어떤 조건이나 목적함수를 만족하는 결과를 얻으려는 방향으로 연구되고 있다. 복잡한 모양이나 구조 등에 고루 이용 가능한 알고리즘의 일반성에 관한 측면에 있어서는 아직은 미진하다.

삼각분할 관련 연구들을 살펴 보면 여러 응용영역 및 계산기하학 자체에의 영향에도 독립된 분야로 취급되지 못했기 때문에 전반적인 연구는 미흡하다. 분할 알고리즘 및 분할된 삼각형의 특성 연구 이외에 문헌 연구로는 특정 삼각분할 부류, 즉 Delaunay나 Greedy 삼각분할에 관한 연구 혹은 2차원 점집합에서 최적 삼각분할에 관해 다룬 연구가 일부 있고[81], 그 이외에는 거의 없다. 따라서 이글에서는 지금까지의 삼각분할 관련 연구에 대해 전반적인 흐름을 정리하고, 앞으로의 연구대상을 종합하여 연구방향을 제시하는 것을 그 목적으로 한다.

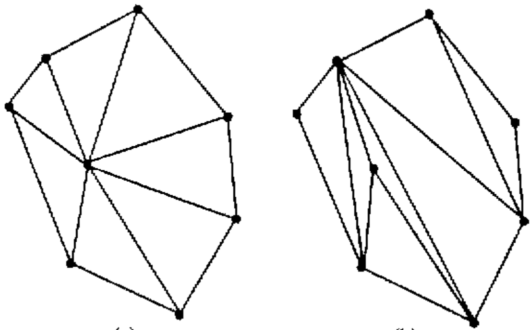
이 글은 전체 5장으로 구성된다. 먼저 II장에서는 2차원 평면에서의 삼각분할을 점, 다각형, 선분 등을 대상으로 주요 특성 및 계산복잡도에 따른 기법들을 살펴보고, III장에서는 3차원 이상의 삼각분할을 II장과 같은 방식으로 다루고, IV장에서는 그 응용분야를 몇 가지 언급한 후, V장에서 결론을 맺는다.

II. 평면에서의 삼각분할

2차원 평면에서의 삼각분할의 대상은 주로 점, 선, 다각형 혹은 이들의 혼합된 집합이다. 삼각분할의 결과가 평면 그래프(Planar graph)이므로 Euler 공식을 만족하는 등 평면 그래프의 특성을 가진다는 사실로부터 여러가지 삼각분할의 특성이 연구 증명되었다.

2.1 점집합의 경우

평면에서 n개의 점집합 S의 삼각분할 T(S)는 T의 어느 두 선분도 S에 속한 점 이외의 다른 점에서 교차하지 않고, 이 선분들이 S의 볼록헐 내부를 삼각형으로 나누는 T의 선분의 집합을 말한다. 점의 갯수와 점의 분포에 따라 여러가지 형태의 삼각분할이 가능하지만, 모든 삼각분할은 공통적으로 S의 점의 갯수를 n, S의 볼록헐 상에 있는 점의 수를 h라 할 때 3n-h-3개의 선분과 2n-h-2개의 삼각형으로 이루어진다[34]. 평면 점집합을 삼각분할하는 방법으로 평면일소(Plane sweep)에 의한 방법이 있다. 이것은 점들을 x좌표값에 따라 왼쪽에서 오른쪽으로 정렬한 후, 처음 세 점으로 삼각분할을 초기화하고 나머지 점들에 대해 현재 처리하는 점에서 보이는 모든 점을 연결하도록 선분을 더해간다. 이것은 정렬에



(그림 2) 같은 집합에 대한 두 가지 삼각분할의 예 (n=8)

의해 O(nlogn)의 시간복잡도를 가지며 점집합 삼각분할의 하한선(Lower bound)과 일치한다.

같은 수의 선분과 같은 수의 삼각형을 가지는 경우도 어떤 선분을 선택하느냐에 따라 여러가지 다른 삼각분할이 가능하다(그림 2). 따라서 여러 삼각분할 중에서 어떤 성질을 만족하거나 목적함수를 최적화하는 삼각분할을 찾는 요구의 문제도 있다. 삼각분할의 최적화 조건으로 제시될 수 있는 것에는 (1) 삼각형의 세각 중 최소각을 최대화(Maxmin angle)[55,66,70,73] (2) 삼각형의 세 각 중 최대각을 최소화(Minmax angle)[38] (3) 각 삼각형을 둘러싸는 가장 작은 원들 중 가장 큰 것을 최소화(Minmax smallest enclosing circle)[73] (4) 최대 외접원의 반지름을 최소화(Minmax circumscribed circle) (5) 삼각분할 선분 중 가장 긴 것을 최소화(Minmax length)[12,37] (6) 삼각분할 선분의 길이의 합을 최소화(Minimum length)[70] (7) 최소 높이의 최대화[8] 등이 있다.

최적화 조건을 만족하도록 삼각분할하는 접근방법으로 Greedy 방법과 Delaunay 방법이 있다[44]. 전자는 요구된 수의 선분이 더해질 때까지 한 번에 하나씩 임의의 삼각분할 선분을 첨가해 가는 방법이다. 후자는 어떤 삼각형의 외접원 내에 다른 점이 놓이지 않는 삼각형들로 구성된다. 이외에도 점집합을 삼각분할하는 데는 선분교환(Edge flip)을 이용한 반복법, 선분 삽입법, 동적 프로레밍 등이 사용된다.

삼각분할을 구성하는 선분의 길이의 합이 최소인 삼각분할(Minimum weight triangulation)을 최적 삼각분할(Optimal triangulation)이라 한다. 일반적인 접근법에 의해 최적 삼각분할을 구하는 다항 시간(Polynomial time) 알고리즘은 아직까지 알려져 있지 않고 이것이 NP-hard인지도 알려져 있지 않다. 그러나, Greedy 삼각분할(GT)과 Delaunay 삼각분할(DT) 등 기존의 방법들이 최적 삼각분할에 얼마나 가까운 삼각분할을 구성하는가에 대한 연구[52,53,56,57,59,63]와, 되도록 최적에 가까운 삼각분할을 구하는 직관적인 알고리즘[74]에 대한 연구가 있다. 평균 시간 복잡도면에서는 일양분포(Uniform distribution)를 가정했을 때, Delaunay 삼각분할과 Greedy 삼각분할이 최적 삼각분할의 길이의 O(logn)배를 가진다[59]. 최악의 경우에는 Manacher와 Zobrist[63]는 임의의 큰 수 n에 대해 최적의 $\Omega(n^{1/3})$배의 길이를 가지는 Delaunay 삼각분할을 만드는 점집합 S′과 최적의 Ω(n/logn)배의 길이를 가지는 Greedy 삼각분할을 만드는 점집합 S″이 있음을 보임으로써 어느 것도 최적 삼각분할을 근사하지 못함을 증명했다. 이와 같이 기존의 방법들이 최적 삼각분할을 근사하지 못하고, 임의의 삼각분할 알고리즘에 대해 최적 삼각분할의 θ(n)배의 길이를 가지는 점집합이 가능하므로[53], 이보다 작은 비율로 최적에 가까운 삼각분할을 구성하는 직관적인 알고리즘이 요구되었다. 집합을 볼록 다각형들로 나눈 후, 각 다각형들에 대해 경계상에서 인접한 점 다음의 점을 연결하는 선분을 반복적으로 첨가해가는 ring heuristic을 적용한 방법에 의한 삼각분할은 최적삼각분할의 O(logn)배의 길이를 가진다[74]. 이 알고리즘의 O(logn)이 오랫동안 최적알고리즘에 가장 가까운 삼각분할의 길이비로 알려졌으나, 최근 가상점을 경계상에만 허용할 때, 사분트리 삼각분할 방법으로 최적 삼각분할의 O(1)배의 길이를 가지는 삼각분할을 구성할 수 있음이 증명되었다[54].

최적 삼각분할이 네트워크 분야 등에서 최소비용의 삼각분할을 구성할 수 있도록 하지만, 때로는 각(angle) 조건 등에 대한 비율을 이용하면 보간 등을 위해서 더 나은 삼각분할을 구성할 수 있다[81]. 각 조건을 최적화 조건으로 하는 경우, Sibson[73]은 Delaunay 삼각분할이 지역 등각(Locally equiangular)인 삼각분할임을 보였고, chew[20]는 좋은 Delaunay 삼각분할이 얻어질 때까지 새로운 점을 추가해 가는 방법을 사용하여 모든 둔각 삼각형이 30°~120°의 각을 갖고 어떤 상수 d에 대해 d부터 2d 사이의 선분 길이를 갖는 삼각분할을 제시했다. S의 삼각분할 T, T′의 m개의 삼각형들의 각 α_i, α'_i들을 크기가 커지는 순서로 나열한 수열($\alpha'_1, \alpha'_2, \cdots, \alpha'_j, \cdots, \alpha'_{3m}$), ($\alpha_1, \alpha_2, \cdots, \alpha_j, \cdots, \alpha_{3m}$)에서 j보다 작은 갯수의 각들이 서로 같고 α'_j가 α_j보다 작은 j가 존재할 때 S의 모든 삼각분할 T′에 대해 삼각분할 T를 전등각(Globally equiangular)이라 한다[33]. Mount와 Saalfeld[66]는 한 원주상에 네 점을 허용하는 점집합에 대해 전등각(Globally equiangular) 삼각분할을 구하는 알고리즘을 제시했다. 이것은 기존의 $O(n^2)$ 보다 나은 O(nlogn) 시간복잡도를 가진다. 이 알고리즘은 각을 증가순으로 배열한 경우 뿐 아니라, 감소순으로 배열한 경우나 선분에 대해서도 이용될 수 있다.

Delaunay 삼각분할은 내부 선분에 대해 마주보는 각의 합이 π보다 크고, 최소각을 최대화하고, 세 점을 지나는 원 내에 다른 점을 포함하지 않는 특성(Empty circle property)을 가진다. Lawson[33]은 Delaunay 삼각분할을 구성하기 위해 선분교환이라는 지엽적 변환에 기초한 방법을 사용하였는데, 이는 임의의 삼각분할을 구성한 후, 연속적인 선분교환으로 Delaunay 삼각분할로 변화시켜가는 방법을 사용했다. Delaunay 삼각분할의 쌍대(Dual) 그레프가 Voronoi 다이얼그램(VD)이므로 이들이 서로 선형시간에 하나에서 다른 하나를 구할 수 있다는 사실을 이용하여 Voronoi 다이어그램으로부터 구하는 방법도 있다[46]. 이는 VD의 복잡도에 따라 Delaunay 삼각분할의 복잡도가 좌우된다. VD를 거치지 않고 바로 구하는 방법으로 보통 분할 정복(Divide-and-conquer) 기법과 반복법(Iterative method)이 사용된다. Shamos[70]는 최초의 분할 정복 알고리즘을 제시하였고 O(nlogn) 시간이 최악의 경우 최적임을 보였다. Lee와 Schachter[55]는 Delaunay 삼각분할과 볼록헐을 사용한 O(nlogn) 시간의 분할 정복 기법과 bin partition을 이용한 $O(n^2)$ 시간의 반복법을 소개했다. Dwyer[32]는 분할 정복에 의해 Delaunay 삼각분할을 구성하는 알고리즘의 평균시간 복잡도를 O(nloglogn)으로 향상시켰다. 이것은 Guibas와 Stolifi[79]가 Voronoi 다이어그램 구성의 중간단계로 Delaunay 삼각분할을 구성했던 방법을 수정하였다. Delaunay 삼각분할의 경우, 선분의 삽입, 삭제를 허용하는 동적인 구조의 유지가 여러 응용 분야에 많이 이용되며[24], 기타 Traveling salesman cycle 혹은 Hamiltonian 경로 등과 관련시켜 Delaunay 삼각분할의 다른 특성에 대한 연구가 있었고[30,28,29,31], 최근에는 일반적인' 위치(General position)에 있지 않는 점들에 대해 오차를 고려하여 Delaunay 삼각분할을 구성할 때, $O(n^2)$의 대각선 교환 알고리즘과 반복 알고리즘의 각 단계에 대해 그 정확성을 보였다[30].

최적 삼각분할을 구하는 알고리즘과 관련하여 삼각분할 선분들 중 가장 긴 선분의 길이를 최소화하는 삼각분할(Minmax length triangulation: MLT)을 찾는 방법이 연구되었다[37]. 이것은 MLT와 Delaunay 삼각분할, relative neighborhood graph(RNG), 볼록헐 사이의 포함관계를 이용하여 볼록헐과 RNG를 구하고 이 둘의 합집합으로 이루어지는 다각형 영역을 구한 후, 동적 프로그래밍 기법에 의해 최대 선분길이를 최소화하는 삼각분할을 구한다.

점집합의 병렬 알고리즘은 CREW PRAM 모델에서 O(n) 처리기로 O(logn) 수행시간을 갖는 최적 알고리즘이 제시되었다[78],

2.2 다각형의 경우

다각형의 삼각분할도 다각형의 모양 및 크기, 구멍(hole) 여부에 따라 여러가지 다른 구성을 가질 수 있다. 그러나, 기본적으로는 n개의 정점을 갖는 단순 다각형 P의 삼각분할은 다각형 내부가 n-2개의 삼각형으로 분할되는데, 이는 서로 교차하지 않는 n-3개의 내부 대각선으로 이루어진다. 다각형 P를 삼각분할하는 단순 알고리즘은 위의 정의로부터 얻을 수 있는데, 주어진 대각선이 다각형 내부에 있는가를 O(n) 시간에 검사하여 대각선의 수가 n-3개가 될 때까지 계속하는 것으로 각 대각선에 대하여 교차검사를 해야 하므로 $O(n^2)$의 시간 복잡도를 갖는다[67].

단순 다각형을 $O(n^2)$ 시간 이내에 삼각분할하는 알고리즘은 1978년에 Garey, Johnson, Preparata와 Shamos[43]가 O(nlogn) 시간의 알고리즘을 처음으로 제시했다. 그들은 이 알고리즘에서 다각형을 정규화하고 단조 다각형들로 나눈 후, 각 단조 다각형에 대해 삼각분할 알고리즘을 적용했다. 분할에 O(nlogn) 시간이 소요되고 단조 다각형 분할은 O(n) 시간에 가능하여 전체 O(nlogn) 시간이 걸린다. 이후 오랫동안 단순 다각형의 삼각분할이 정렬만큼 어렵다고 인식되어 왔는데, 1988년 Tarjan과 Van Wyk[74]가 O(nloglogn) 시간의 알고리즘을 발표함으로써 그것이 정렬(Sorting)의 복잡도 이내에 해결될 수 있다는 것을 증명하였고, 선형시간의 알고리즘 존재 가능성을 높여주었다. 이 방법을 좀 더 자세히 살펴보면, 이것은 삼각분할문제를 가시성의 계산문제로 축소하여 다각형 P를 가시성 계산과 Jordan 정렬을 이용하여 사다리꼴로 분할한 후 각 사다리꼴에 대해 선형시간 삼각분할 알고리즘을 적용한다. 이 알고리즘은 O(nloglogn) 시간을 위해 "finger search tree"라는 복잡한 자료구조의 사용을 요구하며, 효과적인 자료구조의 이용으로 시간 복잡도를 낮춘 예이다. 이에 대해 Kirkpatrick, Klaw와 Tarjan[53]은 좀더 간단한 자료구조를 이용하는 O(nloglogn) 시간의 알고리즘을 제시했다.

O(nlogn) 시간의 최초의 알고리즘에서 O(nloglogn) 시간의 알고리즘이 나오기까지는 두 가지 방향으로 연

구가 이루어졌는데, 첫번째는 특정 다각형 부류에 대해 O(n)의 시간복잡도를 갖는 알고리즘을 찾으려는 시도로, 별모양 다각형, 단조 다각형에 대해 선형시간복잡도의 알고리즘이 제안되었고, 볼록 다각형, 나선형(Spiral) 다각형, 선분가시(Edge-visible) 다각형, 체인 가시(Chain-visible) 다각형 등을 포함하는 선형시간 복잡도를 갖는 다각형 부류와 그에 대한 알고리즘이 제안되었다[43,75,82]. 두번째는 O(nlogn) 시간 복잡도에서 O(n+klogk) 혹은 O(nlogk)로 인수 k를 줄임으로써 복잡도를 선형 수준으로 낮추려 한 것으로, Hertel과 Mehlhorn[47,48]은 오목정점(Reflex vertex)의 수 r에 의한 O(n+rlogr) 시간의 알고리즘을 제시했고, Chazelle과 Incerpi[17]는 다각형의 모양에 따르는 와선도(Sinuosity)라는 개념을 도입하여 와선도가 작을 경우, 그 정점 수에 거의 비례하는 첫 알고리즘을 제시했다.

이후에 Chazelle[16]은 이에 대한 선형시간의 알고리즘을 제시함으로써 이 문제에 대한 하한선이 $\Omega(n)$임을 보였다. 1990년에 발표된 그의 알고리즘은 복잡한 자료구조를 사용하지 않고 두 단계로 나누어 상향 단계에서는 삼각분할의 근사치(Approximation)를 구성하고, 하향 단계에서는 그 삼각분할을 세분화하기 위해 계산된 정보를 이용한다. 여기서는 균형화(Balancing)를 위해 polygon cutting theorem[13]과 새로운 대각선을 찾기 위해 planar separator theorem[61]을 이용한다.

기타 단순 다각형을 분할하는 알고리즘은 다각형을 사다리꼴 혹은 수직 영역들로 분할하는 방법, 동적 프로그래밍 기법을 이용하는 방법들이 있고, 평균시간 복잡도의 경우에 있어서는 Las Vegas 방법을 이용하는 O(nlog*n) 시간의 분할정복 알고리즘이 있다[3,22,40]. 여기서 log*n는 1보다 작은 α에 대해 $(\log n)^{\alpha}$인 값을 말한다. 대개의 경우, 다각형의 삼각분할은 보다 단순한 모양들로 나누어 각각에 대해 처리한 후 병합하거나, 가시맵(Visibility map)을 구성한 후 Jordan 정렬 등의 기법들을 적용하는 방법을 사용한다.

위의 알고리즘들이 다각형 자체의 점 이외에 다른 점, 즉 가상점을 허용하지 않는데 반해, 효율적인 분할을 위해 가상점이 다각형 내부 혹은 그 선분상에 사용될 수 있다. 이것은 점집합의 삼각분할에서처럼 여러가지 최적화 조건을 사용하여 그 조건에 맞는 삼각분할을 구하려 할 때 이용된다. 다각형에서 좋은 삼각분할을 위한 조건이 되는 것으로 둔각이 없는 것, 너무 작은 각이 없는 것, 만들어지는 삼각형의 수가 적당한 것 등이 있다. 삼각형의 세 각 중 가장 큰 각을 최소화하는 삼각분할은 다각형의 외곽을 이루는 선분이 미리 주어져 있다고 생각하여 Deluany 삼각분할의 제한된 경우(Constrained Delaunay 삼각분할)로 해결할 수 있다[20]. 격자 중복법(Grid overlay)에 의해 좋은 삼각분할을 구하는 방법으로는 Baker 등[7]과 Dey[25]의 방법이 있다. 전자는 13°에서 90° 사이의 각을 갖는 삼각분할을 구성할 수 있으나 복잡해서 삼각형의 수에 대해 분석할 수 없고 비다항(Nonpolynomial) 시간복잡도를 가진다. 후자는 12°에서 101° 사이의 각을 가지며 선분과 격자와의 교차의 형태에 따라 경계 혹은 내부에 점을 첨가하여 삼각형의 수를 점의 수와 격자 크기에 의존하는 크기로 제한할 수 있다. Dey[25]는 또한 Delaunay 삼각분할에서 둔각을 갖는 삼각형의 삼각형 중심에 점을 추가하여 삼각분할을 재조정함으로써, 내부삼각형이 30°에서 120° 사이의 각을 가지고 외곽에 대해서는 38.9°에서 97.2°로 제한되는 삼각분할을 제시했다. Baker 등[7]의 방법처럼 둔각을 가지지 않으면서 삼각형 갯수에 대한 분석이 가능한 다항시간 알고리즘으로 Bern과 Eppstein[9]의 방법이 있다. 이 방법은 다각형 내부를 정점을 지나는 수직선으로 나눈 후 다시 정점을 지나는 수평선으로 나누어 간단한 모양들의 집합으로 바꾼 후 사각형 모양일 경우에는 적당한 알고리즘을 이용하여 쉽게 분할하고 둔각삼각형일 경우에는 점을 추가하여 삼각형을 잘라낸 후 분할점의 수를 줄여가는 방법으로 단순다각형, 볼록다각형의 삼각분할 및 임의의 삼각분할의 재구성에 이용한다. 이외는 다소 다르게 Bern 등[10]은 사분트리 방법을 사용하여 다각형을 제한된 종횡비(Aspect ratio)를 갖는 삼각형들 혹은 둔각을 갖지 않는 삼각형들로 분할했다. 삼각형에서 종횡비는 삼각형의 가장 긴 변을 그 변에서 올린 높이로 나눈 비율 또는 삼각형의 가장 긴 변을 가장 짧은 변으로 나눈 비율을 말한다. 이 방법은 최적 삼각분할의 상수배 이내의 길이를 가지고, 입력이 n이고 출력이 k일 때 O(nlogn+k) 시간 복잡도를 가지는, 모양과 크기 모두를 보장하는 방법이다. 이 외에도 최대각을 최소화하는 삼각분할[38]과 최근에는 O(n)개의 삼각형들을 가지며 삼각형의 최소 높이를 가능한 한 최대화하는 삼각분할[8]도 제시되었다.

다각형들이 구멍을 가질 경우, 삼각분할 후 삼각형의 수는 구멍의 수에 따라 달라진다. 즉, h개 구멍, n개 정점을 갖는 다각형은 n+2h-2개의 삼각형으로 분할된다. 구멍을 가지는 경우에 있어서 알고리즘의 복잡도는 구멍의 수 뿐 아니라, 구멍의 모양에 따라 좌우된다. Asano 등[3]은 최소 갯수를 갖도록 다각형을 동적 프로그래밍 방법으로 $O(n^{3+2h})$ 시간에 삼각분할하는 알고리즘을 개발했다. 최근에는 Baker 기법과 사분트리 기

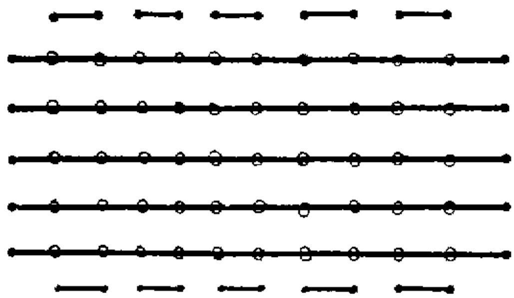
법을 함께 사용하여 최소 갯수의 개념에 작은 각과 둔각을 모두 갖지 않는 삼각분할을 구성하는 알고리즘도 였다[64].

다각형을 삼각분할하는 병렬(Parallel) 알고리즘으로는, 단조 다각형에 대해서는 총수행시간이 선형인 알고리즘이 제시되어 있으나, 단순 다각형의 경우, O(n)개 처리기로 O(logn) 수행시간을 가지는 것이 현재까지 가장 좋은 방법으로 총 O(nlogn)의 수행시간이 걸리지만 O(n)인 순차적 알고리즘에 대해 최적 알고리즘이라고 할 수 없다[79].

2.3 제한된 삼각분할의 경우

점집합이나 다각형과 달리 실제 응용에서 점집합의 삼각분할에 미리 명시된 선분들이 포함되도록 요구하는 경우가 있다. 이처럼 삼각형 선분 집합의 일부가 미리 정의되는 점집합의 삼각분할을 제한된 삼각분할(Constrained triangulation)이라고 한다. 대표적인 것이 제한된 Delaunay 삼각분할(Constrained Delaunay Triangulation: CDT)로, 포함될 선분 집합 일부가 미리 명시되고, 가능한 한 Delaunay 조건을 만족하는 삼각분할이다. 즉, 삼각형의 세 정점 모두에서 볼 수 있는 점을 외접원 내부에 포함하지 않도록 한다. CDT에 대해서는 O(nlogn) 시간 복잡도가 하한선으로 알려져 있다. Lee와 Lin[54]이 $O(n^2)$ 시간의 알고리즘을 제시했고, 1989년 Chew[19]는 분할 정복 기법을 이용한 O(nlogn) 시간의 알고리즘을 발표했다. 이들이 입력점의 수에 대해 가장 좋은 알고리즘을 찾고자 하는 시도였다면 만들어지는 삼각형의 수에 비례하는 시간을 갖는 방법도 있는데, 등변삼각형의 격자 중복(Grid overlay) 후 깊이 우선 탐색으로 점을 더해서 이미 있는 점들에 연결하고 자료점에 일정거리 h보다 가까우면 원의 중심점을 더해가는 Chew의 방법은 분할 후 삼각형의 수가 k일 때 O(k) 시간이 걸린다.

CDT와 같이 점집합의 삼각분할에 선분집합의 일부가 미리 정의되는 것이 아니라, 점집합, 다각형처럼 선분 집합 자체가 삼각분할의 대상이 되는 경우가 있다. 이렇게 선분들의 끝점을 정점으로 하여 선분 집합을 포함하여 삼각분할하는 것을 경계를 가지는(Bounded) Delaunay 삼각분할로 정의할 수 있다. Lee와 Lin이 $O(n^4)$ 시간의 알고리즘을 제시한 바 있고, 이 상호[1]는 Hertel과 Mehlhorn[51]이 단순다각형의 삼각분할에 이용했던 평면일소의 알고리즘과 자료구조를 이용한 O(nlogn)시간 복잡도의 알고리즘과, 분할된 각 부분이 일정갯수의 끝점을 포함하도록 수직분할한 후 선분들을 그



(그림 3) 일치(conform) Delaunay 삼각분할 구성시 첨가되는 점(⊖)

형태에 따라 처리하여 병합하는 $O(n^3)$의 분할 정복 기법 알고리즘을 제시했다. Wang과 Shubert[77]는 Delaunay 삼각분할와 Voronoi 다이어그램의 쌍대성(Duality)을 이용하여 경계를 가지는 Voronoi 다이어그램과 표준(Standard) Voronoi 다이어그램의 차이부분을 교체하는 방법으로 삼각분할을 O(nlogn) 시간에 구성했다.

Delaunay 삼각분할에서 선분을 이루는 두 점을 지나는 원 외부에 다른 모든 점이 놓이도록 두 점을 이어주는데, 넷 또는 그 이상의 점들이 같은 원주 상에 놓이지 않는다는 가정을 할 경우 이런 선분의 모임이 삼각분할을 이룬다면, 원주상의 점을 허용하는 경우에는 각 면은 볼록다각형이 된다. 따라서 이것이 삼각분할이 되기 위해서는 선분을 추가해야 한다. 이때 각 선분에 대해 다른 모든 점들이 그 선분의 끝점을 지나는 원주 상에 있거나 외부에 있게 된다(Empty open disk property). 어떤 그래프 G에 대해 G의 모든 선분들이 위와 같이 구성되는 삼각분할 C의 선분들의 합집합으로 이루어질 경우 G에 대해 일치한다(Conform)고 한다. 임의의 그래프 G에 대해 첨가된 점들의 Delaunay 삼각분할이 G에 대해 일치하는 위와 같은 삼각분할을 가지기 위해서는 몇 개의 점이 첨가되어야 하는가가 하나의 문제가 될 수 있다. 임의의 선분 상의 두 연속점 사이를 간격(Interval)이라 할 때, 각 간격이 모든 다른 선분을 피하는 원을 가지도록 충분히 많은 점을 선분 상에 놓고, 원이 덮는 부분을 제외한 나머지에 대해 반복한다(그림 3). 이때 더해지는 점의 수를 g(n)이라고 할 때, g(n)의 하한선은 Ω(mn)으로 알려져 있는데, 현재까지 알려진 가장 좋은 알고리즘은 $O(m^2n)$이다[36].

임의의 삼각분할에서 어떤 조건을 만족하도록 그것을 개선하는 여러가지 방법이 알려져 있다[41,42]. Frey[42]는 두 가지 개선책을 제시했는데, 구조의 변화없이 위치만을 변경하여 삼각분할을 평활하게 만드는 방법과

공유되는 선분이 일정한 조건을 만족하면 삭제하고 상대쌍을 연결하여 새로운 선분으로 대체하는 선분교환 기법을 이용하여 보다 좋은 특성을 갖도록 했다. 여기서는 또한 각 점이 가질 수 있는 차수값이 좋은 삼각분할을 위한 조건으로 고려된다. Chew[20]의 알고리즘도 제한된 Delaunay 삼각분할에 기초하여 선분을 바꿔가는 기법으로 삼각분할 개선의 한 방법이다.

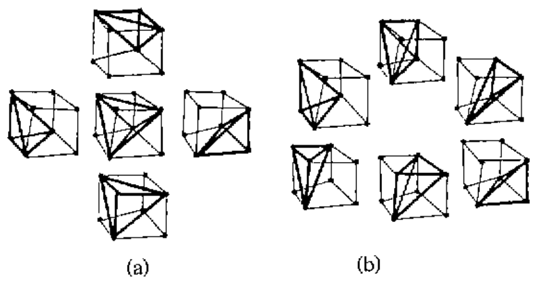
III. 3차원 이상의 삼각분할

삼각분할을 보다 넓은 의미로 정의한다면, 2차원의 삼각형을 3차원에서는 사면체로, 그 이상에서는 그 차원의 단순체(Simplex)로 대체한 단순체들의 복합체로 생각할 수 있다. 고차원에서의 연구는 2차원에서의 결과들을 확장하려는 노력으로 시작되었으나 2차원에서 사용된 방법들이 그 이상의 차원으로 일반화되지 못하는 경우가 많고 고차원 자체의 특성으로 인해 연구된 것도 적다. 고차원에서의 삼각분할을 3차원을 중심으로 살펴보겠다.

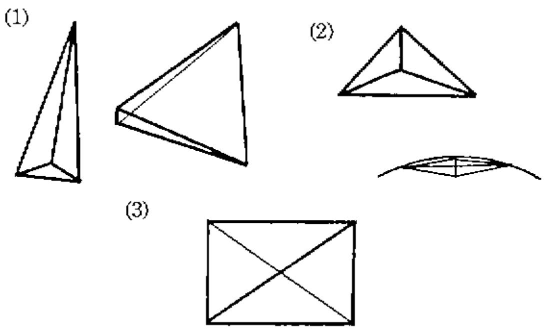
3.1 점집합의 경우

유클리드 공간 ε의 유한 점집합 S의 삼각분할은 단순체(Simplex)의 합집합이 S의 볼록 헐이고, 단순체의 각 정점들의 합집합이 S인 단순체의 모임이다. 공간일소(Space sweep) 방법에 의한 분할은 $O(n^2)$개까지의 단순체를 가질 수 있다. 2차원과 달리 고차원에서는 같은 수, 같은 분포를 갖는 동일 점집합에 대해서도 삼각분할의 방법에 따라 다른 갯수의 단순체로 분할될 수 있다(그림 4). 3차원의 경우 그 갯수의 범위는 O(n)에서 $O(n^2)$까지 다양하다[5]. 포함되는 사면체의 수가 많을 경우, 기억공간 및 저장, 처리 등에 더 많은 노력이 필요하므로, 적은 수의 단순체를 포함하도록 분할하는 것이 바람직하다.

d차원의 경우에는 일반적인 점집합에 대해 적은 갯수의 사면체를 포함하는 삼각분할을 구하는 것이 어렵기 때문에, 어떤 특성을 가지는 집합에 대한 연구로 시작하여 일반화하려는 연구가 있었다. simplical point set은 d차원에서 그 볼록 헐이 d+1개의 정점을 가지는 집합으로, 이 집합이 볼록 헐 내부에 n개 점을 포함할 경우, 각 단순체가 dn/(d+1)개 이하의 점을 포함하는 d+1개 단순체로 분할될 수 있다. 이 때 분할점인 splitter는 $O(d^4 n)$시간에 찾을 수 있고, splitter를 찾는 것에 기초한 분할정복 기법으로 d차원 simplical point set을 삼각분할할 수 있다[5]. 또한 이 방법으로 3차원에서는 임의의 점집합에 대해 O(nlogn+k) 시간에 k개의 단순체로 분할 가능하다.



(그림 4) 육면체를 이루는 6개 점들의 두 가지 분할방법



(그림 5) (1) $\omega=0(1)$, $\kappa\gg1$ (2) $\omega=0(1)$, $\kappa=0(1)$ (3) $\omega\gg1$

분할을 기본으로 한 일반적인 응용에서 구성요소의 수와 함께 중요시 되는 것은 모양이다. 3차원 점집합의 삼각분할은 항상 가능하지만, 분할 후 구성요소인 사면체의 모양이 응용에 바람직하지 못한 경우가 있다. 고차원 삼각분할에서는 사면체의 모양을 정사면체에 가깝도록 하는 것이 균등한 분할을 구성하므로 바람직하고, 특히 네 점이 거의 같은 면에 놓이고 입체각(Solid angle)이 거의 0인 사면체(Sliver)를 피하는 것이 좋다. ω를 외접구 반지름 대 가장 긴 선분 길이의 비, κ를 가장 긴 선분 길이 대 가장 짧은 선분 길이의 비라 할 때, (그림 5)에서 제시되는 사면체는 피해야 할 모양들이다. 즉, (1)과 같은 바늘 모양의 사면체나 (2)와 같은 거의 평면적인 사면체, (3)과 같은 네 변이 비슷한 길이를 가지고 외접구의 표면에 가까이 놓이지 않는 사면체는 sliver를 포함하므로 이런 모양의 사면체가 없도록 하는 삼각분할의 구성이 요구된다. 사면체의 모양을 생각할 때에는 변의 길이와 함께 입체각(Solid angle), 이면각(Dihedral angle), 면각(Face angle)의 세 가지 각(Angle)

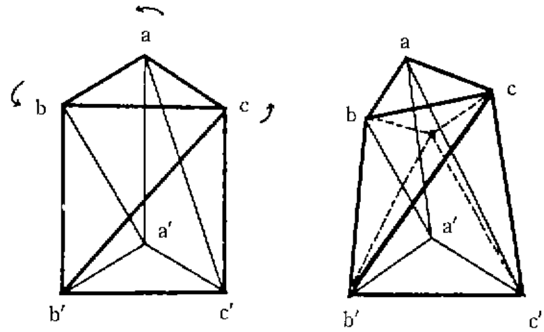
의 조건을 고려해야 하는데, 이들을 다 최적화하는 것이 바람직하지만 그런 일반적인 방법은 아직까지 알려져 있지 않다.

3차원 Delaunay 삼각분할(DT)은 S의 다른 모든 점들이 사면체 t의 네 점의 외접구 밖에 놓이는 사면체 t들로의 분할인데, 이것은 2차원 개념을 확장한 것이지만, 2차원에 해당하는 모든 특성을 만족하지 못한다. 즉, 3차원 DT는 단순체를 포함하는 가장 작은 구의 반지름을 최소화하고, 외접원의 중심이 삼각형 내부에 놓이는 삼각형들을 그 면으로 갖는 사면체들로만 구성되며, flip을 이용한 점진적 알고리즘이 존재한다는 특성을 가지지만, 사면체의 세 종류 각의 어느 것도 최적화되지 못한다. 따라서, 2차원처럼 각 조건만으로 DT를 구별할 수 없으며 구조건이 함께 고려된다. 3차원 Delaunay 삼각분할에 관한 알고리즘은 많이 연구되었는데[4,11,23,35,71, 80], Avis와 Battacharya[4]는 볼록헐의 한 면에서 가장 작은 외접구의 반지름을 가지도록 하는 점을 찾아 사면체를 구성하고 현재 경계면에 대해 반복적으로 사면체를 더해가는 점진적 방법을 사용했고, Watson[80]은 점들을 모두 포함하는 사면체를 구한 후, 내부에 점을 더해 가며 그 점을 포함하는 사면체들을 제거하고 Delaunay 조건을 만족하도록 선분을 교체하는 방법을 사용했다. 이들 알고리즘은 $O(n^3)$의 복잡도를 가진다[11]. d차원 Delaunay 삼각분할의 경우에는 $d+1$ 차원으로 사상(Mapping)하여 변환 처리할 수 있다. 3차원 Delaunay 삼각분할은 2차원의 경우처럼 다른 문제의 준비단계로 많이 이용되고, 그 결과를 임의의 차원으로 확장하는 연구가 활발하다.

각 조건을 최적화하는 삼각분할로 앞에서 언급한 Delaunay 삼각분할, 둔각을 갖지 않는 삼각분할, KJ-삼각분할이 있다[35]. 이들은 사면체의 세 가지 각 중 이면각(Dihedral angle)을 기준으로 한다. KJ-삼각분할은 네 점 a,b,c,d로 이루어지는 T의 선분 ab에 대해 $|cd|$가 cd의 길이이고 ϕ가 cd의 내부각일 때, $-6f(ab)=\Sigma|ab|\cot\phi\geq 0$을 만족하는 삼각분할이다. 이들 세 삼각분할에 대해 다음과 같은 정리가 성립한다.

- 모든 둔각을 갖지 않는 삼각분할은 KJ-삼각분할이지만, 둔각을 갖지 않는 KJ-삼각분할이 존재한다.
- Delaunay 삼각분할이 아닌 둔각을 갖지 않는 삼각분할과 KJ-삼각분할이 아닌 Delaunay 삼각분할이 존재한다.

3차원 삼각분할들은 대개 입력이 어떤 특성을 가지는 집합으로 제한되거나, 좋은 모양을 보장하지 못하거나, 혹은 삼각분할 후 사면체의 수를 어떤 갯수 이내로 제한하지 못하는 경우가 많다. 이들 조건 모두를 함께 고려한 좋은 삼각분할에 대한 연구로는 사면체에서 최소 외접구의 반지름 대 최대 내접구의 반지름인 종횡비(Aspect ratio)를 고려할 때, Bern, Eppstein, Gilbert[10]가 사분트리의 확장이 2^d 트리 삼각분할로 일정 상수 이내의 종횡비를 가지며 최적 삼각분할의 총 선분길이의 상수배 이내의 길이를 갖는 삼각분할을 구성하는 방법을 보였고, Dey, Bajaj, Sugihera[27]는 추가점이 볼록 헐 내부에만 더해진다는 제한 하에서 앞의 (그림 5)의 (1), (2)와 같은 좋지 않은 사면체를 피하는 방법을 제시했다. 위에서 언급한 각조건이나 구조건 혹은 구성 사면체의 수 이외의 다른 최적화 조건으로는 표면적을 최소화한다거나, 체적을 최소화하는 것 등이 고려될 수 있다[21].



(그림 6) 삼각 프리즘의 윗면을 틀어 생긴 다면체

3.2 다면체 및 기타의 경우

다면체의 삼각분할은 다각형의 삼각분할의 자연스러운 확장이다. 그러나 3차원 다면체는 (그림 6)에서와 같이 가상점을 추가하지 않고 삼각분할할 수 없는 경우가 있다[67]. 가상점 여부에 관계없이 다면체를 삼각분할할 수 있느냐를 결정하는 문제도 NP-hard이다[72]. 다면체가 r개의 오목 선분을 가질 경우, $O(r^2)$의 크기로 볼록 분할된다는 사실에서 다면체의 삼각분할을 구성하는 사면체의 갯수의 하한선은 $O(r^2)$임을 알 수 있다. 어떤 다면체에 대한 삼각분할이 가능하지 않을 수도 있기 때문에 다면체의 삼각분할에서는 일반적인 경우보다는 어떤 특성을 가지는 다면체에 대해 주로 연구되었다. Chazelle과 Palios[18]는 구멍과 shell(내부 허공)을 갖지 않는 manifold 다면체를 $O(n+r^2)$ 크기로 삼각분할하는 $O(n+r^2)\log r$) 시간 복잡도의 알고리즘을 제시했는데, 이것은 다면체가 구와 homeomorphic하다는 특성을 이용한다. 구멍과 shell을 허용하는 경우에는 가상점을 추가하여 오목

선분을 제거하는 면으로 볼록이 되도록 반복적으로 자르고 나누어 감으로써 $O(nr^2+r^3 \log r)$의 시간복잡도, $O(nr+r^3)$의 공간복잡도로 삼각분할하는 방법이 제시되었다[26]. Bern, Dobkin, Eppstein[8]은 O(n)개의 구멍을 가지는 거의 편평한 다면체에 대해 크기가 $O(n^2)$이고 각이 150° 이하로 삼각분할하는 방법을 제시했다.

입력을 제한하는 것이 아니라 좋은 특성을 가지는 삼각분할을 만드는 방법으로는 사분 트리의 일반화된 형태를 갖는 특정 격자를 중복시켜 각 원소 내에서 점, 선분, 면과 원소의 경계가 가지는 특성에 따라 삼각분할할 수 있다. Bern, Eppstein, Gilbert[10]의 2^d 트리 방법으로 최적 삼각분할 길이의 상수배 이내를 갖는 삼각분할을 구성할 수 있고, Mitchell과 Vavasis[65]는 구성 사면체들의 종횡비 중 가장 큰 값이 주어진 상수 이하이고, 점의 갯수가 n, 사면체의 수가 m일때, $n=O(m)$인 삼각분할을 하는 방법을 제시했다. 이 방법은 영역의 팔분트리(Octree) 분할에 기초한 것으로 종횡비와 최적성 모두를 고려한 방법이라 할 수 있고, 유한 요소 분석에서 초기 메쉬 생성에 이용된다.

IV. 응 용

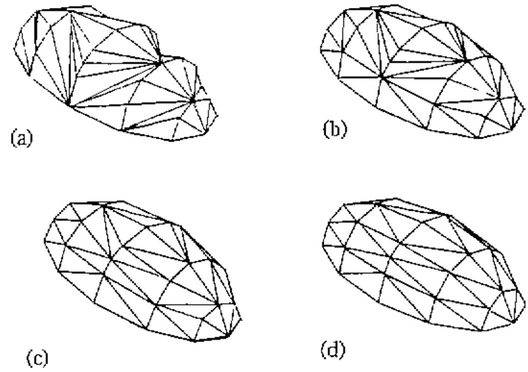
삼각분할은 대상 및 그것이 만족하는 요구조건 등에 따라 여러 분야에서 응용된다. 그러나 앞에서 언급했듯이 그것이 근접문제의 부류인가 분할문제의 부류인가에 따라 나누어질 수 있다.

4.1 근접 문제 부류

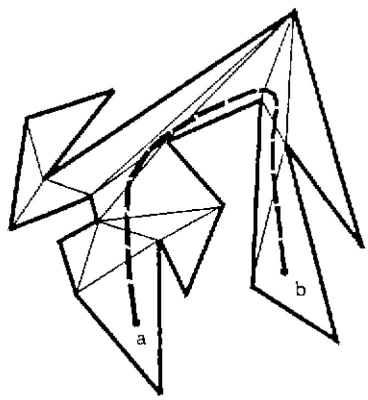
주로 점집합을 그 대상으로 하며 이 부류에 속하는 것으로는 유한 원소 분석, 표면 보간, 모양 형성 등과 가장 가까운 이웃 찾기(Nearest neighbor search) 문제, 점이 포함되는 위치 찾기(Point location) 문제 등이 있다.

유한 원소 분석[12]은 삼각분할문제를 제기한 응용분야이다. 공학 분석(Engineering analysis)에서 부딪히는 많은 문제를 해결하는 수적 절차로 혼합된 경계 조건을 가지는 여러 물질로 구성된 불규칙적인 모양의 물체에 쉽게 적용되고 많은 CAD에서 계산적 기초가 된다. 미적분 등 자료의 근사값을 구하기 위해 자동 메쉬 생성을 요구하는데, 삼각형이 단순한 기하학적 구조로 선호된다. 조건에 맞게 DT를 적용하거나 혹은 밀도 조정 등으로 원하는 삼각분할로 mesh를 생성하여 함수값 보간 등에 이용한다.

표면의 보간은 3차원 자료의 입력시 세번째 좌표값의 범위가 크지 않을 경우, 2차에 대한 삼각분할을 구성함으로써 원하는 면의 굴곡을 보간한다(그림 7). 이것은 그래픽 출력(Graphic display) 및 수치해석의 계산에도 유용하다[21,40,68]. 표면 보간의 경우, 위에서 언급되었던 최적화조건 중 일부를 만족하는 방향으로도 가능하지만, 그 이외에 자료가 가지는 값에 따라 특정한 함수값을 만족하도록 하는 것이 더 적절할 수 있다[33].



(그림 7) 삼각분할을 이용한 표면 보간의 예



(그림 8) 삼각분할된 다각형에서 최단 경로 찾기

4.2 분할 문제 부류

분할 문제의 응용으로 사용될 때 그 대상은 다각형 혹은 다면체이다. 분할은 복잡한 모양의 물체를 단순한 것의 모임으로 간주하고 문제를 풀거나, 나뉘어진 영역에 따라 단계적으로 문제를 해결하는데 이용한다. 계산기하영역에서의 기하학적 분할[40], 가시성(Visibility)[13, 15,45,49,50] 및 최단 경로[45] 문제, 분리성(Separability)[53], 선분로부터 단사질의(Shooting query)문제 등은 주어진 영역을 분할하여 단계적으로 문제를 해결하려

한 일례이다. 이들은 처리의 준비단계로 대상영역이 삼각분할될 것을 요구하는데, 전처리 후에는 선형시간에 문제해결이 가능한 것이 대부분으로 선형 삼각분할 알고리즘이 개발됨에 따라 최적 방법으로 간주된다. 두 점 사이의 최단거리를 구하려 할 경우, 삼각분할에 대응되는 쌍대 그래프를 구한 후, 점이 포함된 삼각형에서 인접한 삼각형으로 선을 연결해 가며 구할 수 있다. (그림 8)은 삼각분할된 그래프에서 두 점 사이의 최단거리를 구하는 것을 보여준다.

Prison yard 문제[67]는 다각형의 삼각분할에서 내부가 아니라 외부에 속하는 블록 헐 영역을 삼각분할하는 것이 유용할 수 있다. 기타, 평면의 삼각분할을 이용한 3차원 모델링이나, CAD, 패턴 인식 등에서도 자료의 처리 및 저장을 쉽게 하기 위해서 분할이 이용될 수 있다.

V. 결 론

앞에서 최근까지 발표된 삼각분할에 관한 연구들을 삼각분할 대상을 중심으로 살펴보았다. 이제까지의 삼각분할 연구는 하나의 대상에 대해 보다 빠르고 효율적인 알고리즘을 구하려는 연구나, 어떤 조건이나 목적함수를 만족하는 결과를 얻음으로써 다른 문제에 대한 보다 효율적인 해결을 도우려는 방향으로 연구되고 있음을 알 수 있다. 또한 분할 정복 또는 동적 프로그래밍 등 기존의 여러 알고리즘 설계 기법들이 삼각분할의 기본적 구조로서, 혹은 그 일부 단계로서 사용될 수 있었다. 이 때, 기존의 방법을 사용하여 수행시간을 낮추려는 노력은 좀 더 복잡한 자료구조 및 알고리즘 단계를 요구한다. 볼록분할[76]이나 점이 포함되는 영역을 찾는 문제 등의 기본적인 문제들은 삼각분할의 중간 부산물로 혹은 주요 단계로서 삼각분할에 영향을 주었다. 삼각분할과 그 중간 단계로서 요구되는 문제들, 응용분야들 사이의 관계로부터 기존의 기법들에 대한 좀 더 빠른 알고리즘의 개발이 삼각분할 문제에 영향을 주고 삼각분할 문제를 이용하는 다른 문제에 대한 효율적인 해결책을 제시한다는 것을 알 수 있다.

이 글에서는 이전의 연구가 2차원 삼각분할에 관련한 것이 많고 3차원 이상의 경우에는 단순한 구조를 가지거나 제한된 특성을 가지는 어떤 특정 집합에 한한 것들이라 2차원에 관해 주로 언급되었다. 또한 일반적인 알고리즘 및 randomized 알고리즘, 병렬(Parallel) 알고리즘에 대해서는 거의 언급하지 못했다. 이 글에서는 삼각분할을 대상별로 나누어 연구결과를 살펴보았는데, 삼각분할의 연구가 최근의 경우, 2차원의 방법을 언급하며 3차원 이상에의 확장성을 논하고 있다는 측면에서 기법 및 조건에 따른 분류조사도 의미있는 작업이 될 것이다.

이전의 연구가 2차원 다각형 등 어떤 한 분야에서는 비교적 완성되었다고 하지만 대부분의 경우 진행중인 단계이다. 3차원 이상의 경우 특히 다면체에 있어서는 입력되는 다면체에 대한 정확한 묘사가 어렵기 때문에 그 특성을 이용하는 효율적인 방법이 아직은 부족하다. 따라서 보다 좋은 특성을 가지는 삼각분할 및 3차원 이상의 분할 대상에 대한 분석과 새로운 기법의 개발이 요구된다. 앞으로 더 연구를 필요로 하는 것으로 다음과 같은 것들이 있다.

(1) 점집합에서 최적 삼각분할을 구하는 구하는 알고리즘이 있는가 혹은 이 문제는 NP-hard인가?

(2) 내부 점만을 더해서 좋은 특성을 갖도록 삼각분할되는 다각형의 특성은?

(3) 삼각분할 후 가장 짧은 선분을 최대화하는 삼각분할을 구하는 문제

(4) Chazelle[16]의 방법보다 간단한 단순다각형의 삼각분할 방법

(5) Hamiltonian circuit과 Delaunay 삼각분할 사이의 관련된 문제

- 대부분의 DT가 Hamiltonian인가?
- 주어진 DT에서 Hamiltonian 결정이 얼마나 어려운가?

(6) Conforming Delaunay 삼각분할에서 상한선 $O(m^2 n)$과 하한선 $O(mn)$ 사이의 차이는 줄일 수 있는가?

(7) [flip distance] 임의의 점집합에 대한 여러 삼각분할의 집합이 주어질 경우, 하나의 선분을 교환함으로써 다른 삼각분할로 바뀔 수 있으며 이를 이용한 삼각분할들의 정렬이 가능한가?

(8) 3차원 점집합 P의 일치 삼각분할 구성에서 Q가 둔각을 갖지 않는 삼각분할을 갖도록 하고자 할 때 추가될 점의 최소수 $g(n)$은 $(P \subseteq Q)$?

(9) KJ-삼각분할에서 예각인 이면각을 가지는 경우, 어떤 상수 α를 넘지 않는 이면각만 가지는 경우 각각 $g(n)$의 값은?

(10) Simplical point set에 적용된 알고리즘이 d차원으로 확장가능한가?

(11) 볼록이 아닌 다면체를 삼각분할하는 가장 좋은 수행시간

(12) 삼각분할하기 쉬운 다면체의 특성 및 그 방법

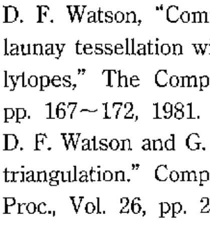
(13) 종횡비 이외에 사면체의 모양에 관해 가능한 특성 및 그에 대한 알고리즘

참 고 문 헌

1. 이상호, "선분들의 삼각분할을 위한 효율적인 알고리즘," 정보과학회지 제6권 제4호 (15-4-9) 1988. 8.
2. 양태천, "단순다각형을 분할하기 위한 개선된 알고리즘," 한국과학기술원 전산학과, 석사학위논문, 1984.
3. T. Asano, T. Asano and R. Y. Pinter, "Polygon triangulation: Efficiency and minimality," J. Algorithms, Vol. 7, pp. 221~231, 1986.
4. D. Avis and B. K. Bhattacharya, "Algorithms for computing d-dimensional Voronoi diagrams and their duals," F. P. Preparata (ed.), Advances in Computing Research, Vol. 1, JAI Press, Greenwich, pp. 159~180, 1983.
5. D. Avis and H. ElGindy, "Triangulating simplical point sets in space," 2th Symp. Computational Geometry, pp. 133~141, 1986.
6. D. Avis and G. T. Toussaint, "An efficient algorithm for decomposing a polygon into star-shaped polygons," Pattern Recogn., Vol. 13, pp. 395~398, 1981.
7. B. S. Baker, E. Grosse and C. S. Rafferty, "Nonobtuse triangulation of polygons," Discrete and Computational Geometry, Vol. 3, pp. 147~168, 1988.
8. M. Bern, D. Dobkin and D. Eppstein, "Triangulating polygons without large angles," 8th Symp. Computational Geometry, pp. 222~231, 1992.
9. M. Bern and D. Eppstein, "Polynomial-size nonobtuse Triangulation of polygons," 7th Symp. Computational Geometry, pp. 357~363, 1991.
10. M. Bern, D. Eppstein and J. Gilbert, "Probably good mesh generation," FOCS, pp. 1~11, 1990.
11. J. D. Boissonnant, O. D. Faugeras and E. Le Bras-Behlman, "Representing stereo data with the Delaunay triangulation," IEEE, 1988.
12. J. Bramble and M. Zlamal, "Triangular elements in the finite element method," Math, Comp., Vol. 24, pp. 809~820, 1970.
13. B. Chazelle, "A theorem on polygon cutting with applications," Proc. 23rd Ann IEEE Symp. on FOCS, pp. 339~349, 1982.
14. B. Chazelle, "How to divide a polygon fairly," Tech. Rept. CMU-CS-82-120. Dept. of Comput. Sci., Carnegie-Mellon Univ., Pittsburgh, Pennsylvania, Apr. 1982.
15. B. Chazelle. "The power of triangulation: Application to problems of visibility and internal distance," Tech. Rept. CMU-CS-82-121, Dept. of Comput. Sci., Carnegie-Mellon Univ., Pittsburgh, Pennsylvania. March. 1982.
16. B. Chazelle, "Triangulating a simple polygon in linear time," Tech. Rept. CS-TR-264-90, Dept. of Comput. Sci., Princeton Univ., 1990.
17. B. Chazelle and J. Incerpi, "Trianulation and shape-complexity," ACM Trans. Graphics. Vol. 3, pp. 135~152, 1984.
18. B. Chazelle and L. Palios, "Triangulating a non-convex polytope," Discrete and Computational Geometry, Vol. 5, pp. 505~526, 1990.
19. L. P. Chew, "Constrained Delaunay triangulation," Symp. Computational Geometry, Vol. 3, pp. 215~222, 1987.
20. L. P. Chew, "Guaranteed-quality triangular meshes," Tech. Rept. TR-89-983, Dept. Comput. Sci., Cornell Univ., Ithaca, NY, 1989.
21. B. K. Choi, H. Y. Shin. Y. I. Yoon and J. W. Lee, "Triangulation of scattered data in 3D space," CAD 20, pp. 239~248, 1988.
22. K. L. Clarkson, R. E. Tarjan and C. J. Van Wyk, "A fast Las Vegas algorithm or triangulating a simple polygon," 4th Symp. Computational Geometry, pp. 18~22, 1988.
23. P. A. Devijver and M. Dekesel, "Computing multidimensional Delaunay tessellations," Pattern Recogn, Lett., Vol. 1, pp. 311~316. 1983.
24. P. A. Devijver and M. Dekesel, "Insert and delete algorithms for maintaining dynamic Delaunay triangulations," Pattern Recogn. Lett., Vol. 1, pp. 73~78, 1982.
25. T. K. Dey, "Good triangulations in plane," Proc. 2nd Canad. Conf. Comput. Geom., pp. 102~106, 1990.
26. T. K. Dey, "Triangulation and CSG representation of polyhedra with arbitrary genus," 7th Symp. Computational Geometry, pp. 364~372, 1992.
27. T. K. Dey, C. Bajaj and K. Sugihara, "On good triangulation in three dimensions," Poc. of Symp. on Solid Modeling Foundations and CAD/CAM Applications(CAD/SIGRAPH), Texas, 5~7, June. 1991.
28. M. B. Dillencourt, "A non-Hamiltonian, nondegenerate Delaunay triangulations," Inform. Process. Lett., Vol. 25, pp. 149~151, 1987.
29. M. B. Dillencourt. "Realizability of Delaunay triangulations," Inform. Process. Lett., Vol. 33, pp. 283~287, 1980/90.

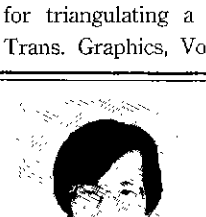
30. M. Dillencourt, "Toughness and Delaunay triangulation," 3th Symp. Computational Geometry, pp. 186~194, 1987.
31. M. B. Dillencourt, "Travelling salesman cycles are not always subgraphs of Delaunay triangulations or of minimum weight triangulations," Inform. Process. Lett., Vol. 24, pp. 339~342, 1987.
32. R. A. Dwyer, "A simple divide-and-conquer algorithm for computing Delaunay triangulations in O (nloglogn) expected time," 2nd Symp. Computational Geometry, pp. 276~284, 1986.
33. N. Dyn, D. Levin and S. Rippa, "Data dependent triangulations for piecewise linear interpolation." IMA Journal of Numerical Analysis, Vol. 10, pp. 137~154, 1990.
34. H. Edelsbrunner, Algorithms in Combinatorial Geometry, Springer-Verlag, Heidelberg, Germany, 1987.
35. H. Edelsbrunner, "Spatial triangulations with dihedral angle conditions," Proc. of Intl. Workshop on Discrete Algorithms and Complexity, Fukuoka, Japan, pp. 83~89, 1989.
36. H. Edelsbrunner and T. S. Tan, "An upper bound for conforming Delaunay triangulations," 8th Symp. Computational Geometry, pp. 53~62, 1992.
37. H. Edelsbrunner and T. S. Tan, "A quadratic time algorithm for the minmax lenghth triangulation," Tech. Rept. UIUCDCS-R-91-1665, Dept. Comput. Sci., Univ. of Illinois, Urbana, Illionois, Feb. 1991.
38. H. Edelsbrunner, T. S. Tan and R. Waupotitsch, "An O(n^2logn) time algorithm for the minmax angle triangulation." Tech. Rept UIUCDCS-R-90-1575, Dept. Comput. Sci., Univ. of Illinois, Urbana, Illinois, 1990.
39. S. Fortune, "Numerical stability of algorithms of for 2D Delaunay triangulations and Voronoi diagrams," 8th Symp. Computational Geometry, pp. 83~92, 1992.
40. A. Fournier and D. Y. Montuno, "Triangulating simple polygons and equivalent problems." ACM Trans. Graphics, Vol. 3, pp. 153~174, 1984.
41. W. H. Frey. "Selective refinement: A new strategy for automatic node placement in graded triangular meshes," Intern. J. for Numerical Methods in Engineering, Vol. 24, pp. 2183~2200, 1987.
42. W. H. Frey and D. A. Field, "Mesh relaxation: A new technique for improving triangulations," Math. Dept., General Motors research laboratories, Warren, MI, Sept. 1989.
43. M. Garey, D. S. Johnson, F. P. Preparata and R. E. Tarjan, "Triangulating a simple polygon," Inform. Process. Lett. Vol. 7, No. 4, pp. 175~180, 1978.
44. S. A. Goldman, "A space efficient greedy triangulation algorithm," Inform. Process. Lett., Vol. 31, pp. 191~196, 1989.
45. L. Guibas, J. Hershburger, D. Leven, M. Sharir and R. E. Tarjan, "Linear time algorithm for visibility and shortest path problems inside triangulated simple polygons," Algorithmica, Vol. 2, pp. 209~233, 1987.
46. L. J. Guibas and J. Stolfi, "Primitives for manipulation of general subdivisions and the computation of Voronoi diagrams," ACM Trans. Graphics, Vol. 4, pp. 74~123, 1985.
47. S. Hertel and K. Mehlhorn, "Fast Triangulation of simple polygons," Proc. 4th Intern. Conf. Found. Comput. Theory. Lecture Notes in Computer Science, Vol. 158. Springer-Verlag, pp. 207~218, 1983.
48. S. Hertel and K. Mehlhorn, "Fast triangulation of the plane with respect to simple polygons," Information and Control, Vol. 64, pp. 307~310, 1987.
49. J. Hershberger, "An optimal visibility graph algorithm for triangulated simple polygons," Algorithmica, Vol. 4, pp. 141~155, 1989.
50. J. Hershberger, "Optimal parallel algorithms for triangulated simple polygons," 8th Symp. Computational Geometry, pp. 33~42, 1992.
51. J. M. Keil, "Decomposing a polygon into simpler components," SIAM. J. Comput., Vol. 14, No. 4, pp. 799~817, 1986.
52. D. G. Kirkpatrick, "A note on Delaunay and optimal triangulations," Inform. Process. Lett. Vol. 10, pp. 127~128, 1990.
53. D. Gl. Kirkpatrick, M. M. Klawe and R. E. Tarjan, "Polygon triangulation in O(nloglogn) time with simple data structures," 6th Symp. Computational Geometry, pp. 34~43, 1990.
54. D. T. Lee and A. K. Lin. "Generalized Delaunay triangulation for planar graphs," Discrete and Computational Geometry, Vol. 1, pp. 201~217, 1986.
55. D. T. Lee and B. Schachter, "Two algorithms for constructing Delaunay triangulations," Int. J. Comput. Inform. Sci., Vol. 9, No. 3, pp. 219~242, June 1980.
56. C. Levcopoulos, "An $\Omega(\sqrt{n})$ lower bound for the nonoptimality of the greedy triangulation," Inform. Process. Lett., Vol. 25, pp. 247~251, 1987.
57. C. Levcopoulos and A. Lingas, "On approximation behavior of the greedy triangulation for convex po-

lygon," Algorithmica, Vol. 2, pp. 175~193, 1987.
58. B. A. Lewis and J. S. Robinson, "Triangulating of planar regions with applications," The Computer Journal, Vol. 21, pp. 324~332.
59. A. Lingas, "The greedy and Delaunay triangulations are not bad in the average case," Inform. Process, Lett., Vol. 22, pp. 25~31, 1986.
60. A Lingas, "On partitioning polygons," Tech. Rept. LITH-IDA-R-85-13, Linkoping Univ., pp. 1~8, 1985.
61. R. J. Lipton and R. E. Tarjan, "A separator theorem for planar graph," SIAM J. Comput., Vol. 36, pp. 177~189, 1979.
62. A. Lubiw, "Decomposing polygonal regions into convex quadrilaterals," 1st Symp. Computational Geometry, pp. 97~106, 1985.
63. G. K. Manacher and A. L. Zobrist, "Neither the greedy nor the Delaunay triangulations of a planar point set approximates the optimal triangulation," Inform. Process. Lett., Vol. 9, pp. 31~34, 1979.
64. E. A. Melisaratos and D. L. Souvaine, "Coping with inconsistencies: a new approach to produce quality triangulations of polygonal domains with holes," 8th Symp. Computational Geometry, pp. 202~211, 1992.
65. S. A. Mitchell and S. A. Vavasis, "Quality mesh generation in three dimensions", 8th Symp. Computational Geometry, pp. 212~221, 1992.
66. D. Mount and A. Saafield, "Globally-equiangular triangulations of co-circular points in O(nlogn) time," 4th Symp. Computational Geometry, pp. 143~152, 1988.
67. J. O'Rouke, Art Gallery Theorems and Algorithms, Oxford University Press, New York, 1987.
68. A. Oxley, "Surface fitting by triangulation," The Computer Journal, Vol. 28, pp. 335~339, 1985.
69. D. A. Plaisted and J. Hong, "A heuristic triangulation algorithm," J. Algorithms, Vol. 8, pp. 405~437, 1987.
70. F. F. Preparata and M. I. Shamos, Computational Geometry: An Introduction, Springer-Verlag, Heidelberg, New York, 1985.
71. V. T. Rajan, "Optimality of the Delaunay trinagulation in R^d," 7th Symp. Computational Geometry, pp. 357~356, 1991.
72. J. Ruppert and R. Seidel, "On difficulty of triangulating three-dimensional nonconvex polyhedra," Discrete and Computational Geometry, Vol. 7, pp. 227~253, 1992.
73. R. Sibson, "Locally equiangular triangulations," The Computer Journal, Vol. 21, pp. 243~245, 1978.
74. R. E Tarjan and C. J. Van Wyk, "An O(nloglogn) time algorithm for triangulating a simple polygons," Tech. Rept. CS-TR-052-86, Princeton Univ., 1986.
75. G. Toussaint, "A new linear algorithm for triangulating monotone polygons," Pattern Recogn. Lett., Vol. 2, pp. 155~158, 1984.
76. G. Toussaint and D. Avis, "On a convex hull algorithm for polygons and its application to triangulation problems," Pattern Recogn., Vol. 15, pp. 23~29, 1982.
77. C. Wang and L. Shubert, "An optimal algorithm for constructing the Delaunay triangulation of a set of line segments," 3th Symp. Computational Geomerty, pp. 223~232, 1987.
78. C. A. Wang and Y. H. Tsin, "An O(logn) time parallel algorithm for triangulating a set of points in the plane," Inform. Process, Lett., Vol. 25, pp. 55~60, 1987.
79. H. Wagner, "Triangulating a monotone polygon in parallel," Computational Geometry and its applications, CG '88 Intern. Workshop on Computational Geometry, Lecture notes in Computer Science, Vol. 333, edited by H. Noltemeier, pp. 136~147, 1988.
80. D. F. Watson, "Computing the n-dimensional Delaunay tessellation with applications to Voronoi polytopes," The Computer Journal, Vol. 24, No. 2, pp. 167~172, 1981.
81. D. F. Watson and G. M. Philip, "Survey-systematic triangulation." Comp. Vision, Graphics and Image Proc., Vol. 26, pp. 217~223, 1984.
82. T. C. Woo and S. Y. Shin, "A linear time algorithm for triangulating a point visible polygon," ACM Trans. Graphics, Vol. 4, No. 1, pp. 60~69, 1985.



이 상 호

1979 서울대학교계산통계학과졸업
1979 ~1987 한국과학기술원 전산학과 이학석사, 공학박사
1990 ~1991 미국 일리노이 대학교 연구교수
1983 ~현재 이화여자대학교 전자계산학과 부교수 전자계산연구소장
관심 분야 : 자료구조 및 알고리즘, 계산기하학, 그래프 이론



신 금 림

1991 이화여자대학교 전자계산학과 졸업(학사)
1991 ~현재 이화여자대학교 전자계산학과 석사 과정
관심 분야 : 자료구조 및 알고리즘, 계산기하학, 그래프 이론, 계산이론 등