한국섬유공학회지, 제43권 제1호 2006년
<연구논문>

Textile Science and Engineering
Vol. 43, No. 1, 2006

3차원 의복 디자인에서의 예각 다트를 고려한 삼각형 메쉬 생성 알고리즘

설인환† · 김성민[1] · 강태진
서울대학교 재료공학부, [1]전남대학교 응용화학부
(2005. 11. 23. 접수/2006. 2. 3. 채택)

Triangular Mesh Generation Algorithm for an Acute Angle Dart in 3D Apparel Design

In Hwan Sul†, Sung Min Kim[1] and Tae Jin Kang
Intelligent Textile System Research Center and School of Materials Science and Engineering, Seoul National University, Seoul 151-744, Korea
[1]Faculty of Applied Chemistry, Chonnam National University, Gwangju 500-757 Korea
(Received November 23, 2005/Accepted February 3, 2006)

Abstract: A new Triangular mesh generation algorithm based on Delaunay triangulation for three-dimensional apparel CAD system was developed. Triangular element shape was chosen to represent complex pattern shapes effectively. The triangulation scheme is based on incremental searching method and extreme pattern geometries such as a narrow dart angle, which conventional mesh generation algorithms could not deal with, was considered. The initial pattern shape is assumed to be a general non-convex polygon with holes and darts. Input data were transformed to an integer domain to reduce round off error. Local triangulation scheme and edge-based half-plane searching scheme were used to optimize calculation speed and the resultant meshes showed good triangulation for 3D apparel CAD patterns.

Keywords: *triangular mesh, delaunay triangulation, dart, 3D apparel CAD*

1. 서 론

현실에서 발생하는 복잡한 문제들을 수치해석적인 방법으로 풀기 위해서는 유한화 (discretization)의 과정이 필요하다. 직물의 드레이프 모사에 대한 연구는 유한요소법으로부터 출발하여, 현재는 스프링-대쉬팟 모델 (spring-dashpot model)과 같은 파티클 법 (particle based method)이 빠른 연산속도 때문에 주로 쓰이고 있다. 그런데 역학분야에서 유한요소법 (FEM)이나 유한차분법 (FDM)으로 주어진 형상을 유한화하는 것과 마찬가지로, 직물의 드레이프 모사를 위해서도 직물을 일정한 단위 형상을 갖는 요소로 변환할 필요가 있다.

파티클 법을 드레이프에 응용함에 있어서도 의복 패턴을 삼각형 또는 사각형의 메쉬로 유한화하는 것이 필요한데, 파티클 법에 있어서의 메쉬 생성은 유한요소법의 그것과 비교해 차이점이 있다. 유한요소법은 메쉬의 최대 또는 최소각이 일정한 범위를 넘어설 경우 결과값의 오차가 커지는 단점[1]이 있기 때문에 기존의 유한요소용 메쉬 생성 알고리즘들은 이러한 내각의 분포를 최적화시키는데 중점을 두어 연구되어 왔다. 파티클 법은 유한요소법에 비해 요소 (element)의 내각에 대한 의존성은 적지만, 사용되는 의복 패턴의 외곽선 형상이 복잡하고 홀이나 다트와 같은 특이한 경계 조건들을 갖기 때문에 이들을 고려하여야 메쉬를 생성하여야 한다. 2차원 메쉬에 있어서 주로 사용되는 삼각형과 사각형 요소 중에서, 삼각형 요소가 충돌검사나 3차원 디스플레이등에서 사용되는 기본적인 형태이므로 삼각형 요소가 더 의복 패턴에 있어서는 적절한 접근방법이라 할 수 있다[2].

그런데 어패럴 캐드에 있어서의 메쉬 생성법이 다른 기존의 캐드법과 다른 점은, 의복은 내부 홀 (hole)과 다트 (dart)의 도입으로 인해 외곽선이 매우 작은 내각을 갖는다는 것이다. 물론 기존의 메쉬 생성법들로도 simply connected boundary에 edge가 첨가된 형상은 메쉬를 생성할 수 있지만, 다트의 내각이 이론적으로 0인 경우 기존의 방법들은 처리하지 못하는 단점이 있다. 다트의 내각이 0이라는 것은 바운더리에 같은 좌표를 갖는 노드들이 두개씩 있거나 같은 선분위에 노드들이 중첩되어 있는 것을 의미하므로, 기존의 캐드 응용 분야에서는 이러한 적용 분야를 찾아보기 힘들기 때문이다. 본 연구에서는 이러한 0도 각도의 다

† Correspondence to In Hwan Sul (snowman0@hitel.net)

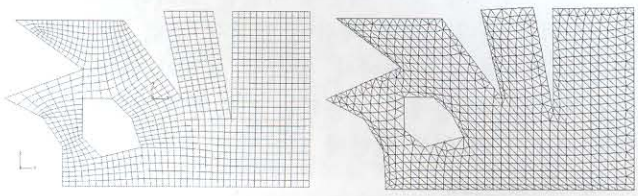
트까지도 처리가능한 삼각형 메쉬 생성 알고리즘을 Delaunay 삼각화법에 기반하여 개발하였고, 실제 패턴 형상에 대입하여 메쉬를 생성하였다.

2. 삼각형 메쉬 생성

2.1. Mesh Element Shape

초기의 드레이프 연구에서는 테이블 보와 같은 간단한 직물에 대해 사각형 요소를 사용하였다. 사각형 요소는 각 노드의 이웃하는 노드 (neighboring nodes)의 수가 4로 고정되어 있어서 알고리즘이 간단하고 직물의 경사-위사 구조와 유사하다는 장점을 갖고 있다는 주장[3]이 있었다. 그러나 테이블 보가 아닌 실제 복잡한 형상의 의복 패턴에 대해 사각 요소를 적용할 경우, *Figure* 1(a)와 같이 요소의 선분 (edge)가 경사 위사와 다른 방향으로 놓이게 되어서 선분마다 경사와 이루는 각에 따라 물성을 대입하는 과정이 필요하다. 즉 사각형 요소를 사용할 때의 장점은 직물 구조 (weave structure)에의 유사성 보다는 메쉬 구조의 단순함으로 인한 편리성이라고 할 수 있다. 사각형 요소로도 패턴의 형상을 나타낼 수 있지만, 3차원에서의 충돌검사나 OpenGL® 등을 통한 디스플레이에 있어서 삼각형이 기본 요소이므로 삼각형 요소를 사용하는 것이 더 편리하다. 삼각형 요소는 이웃노드의 수가 일정하지 않으므로 알고리즘 구현에 있어서 어려움이 있지만, 사각형 요소에 비해 복잡한 형상을 더 잘 나타낼 수 있는 장점이 있다. 따라서 본 연구에서는 삼각형요소를 기반으로 메쉬 생성 알고리즘을 고안하였다.

삼각형 요소 (*Figure* 1b)를 사용함에 있어서도 사각형 요소에서와 같이 요소 선분이 이루는 각에 따라 직물 물성을 대입해주는 것이 필요하다. *Figure* 2에서 직교하는 경사-위사 구조에서 선분 *st*가 경사와 이루는 각을 θ라고 할 때 이 선분이 갖는 비대칭 (anisotropic) 물성 $B_{st}(\theta)$를 경사의 물성(B_w)와 위사의 물성(B_f)의 내삽 (interpolation)으로 나타내는 과정을 보이고 있다.



(a) Quadrilateral (from ABAQUS®) (b) Triangular

Figure 1. Comparison of mesh element shape.

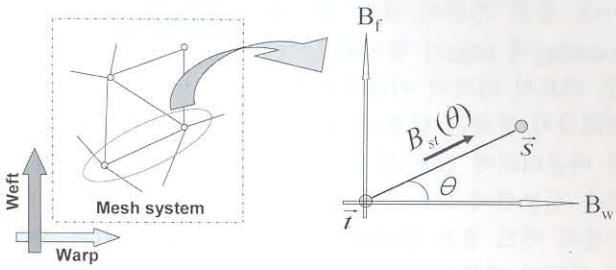


Figure 2. Assignment of anisotropic material property.

2.2. Delaunay 삼각형법

메쉬 바운더리가 지정되고 메쉬 노드가 적절히 분포되어 있다고 가정할 때, 주어진 노드들을 삼각형의 집합으로 변환하는데 있어서 가장 널리 쓰이는 방법이 Delaunay 삼각형법 (triangulation)[4-7]이다. Delaunay 법은 임의의 삼각형 요소가 이루는 외접원 내에 삼각형을 이루는 노드 이외의 노드가 존재하지 않도록 하여 삼각화를 이루는 방식이다. 이 방법을 사용하면 유한요소법에 있어서 이등변 삼각형에 가장 가까운 삼각형 집합을 이루어 냄으로써 가장 오차가 적은 연산 결과[7]를 얻을 수 있다. *Figure* 3에서 Delaunay test를 이용할 때와 이용하지 않을 때의 삼각화 결과를 간단한 경우에 대해서 나타내고 있는데 3(a)의 경우 3(b)에서 보이는 날카로운 내각을 갖는 요소가 없음을 볼 수 있다.

그러나 Delaunay test만으로 삼각화를 할 수는 없는데 *Figure* 4와 같이 동심원상에 노드들이 존재하는 경우 모든 경우 Delaunay test에 적합하지만 최소내각이 큰 경우가 발생할 수 있다. 따라서 삼각형 집합을 검색함에 있어 최소내각을 측정하여 최소내각이 가장 작은 삼각형 집합을 먼저 선택해야 좋은 결과를 얻을 수 있다. 따라서 전체 삼각화 과정은 최소내각을 찾는 첫번째 탐색 과정과, 나머지 찾지 못한 삼각형들을 이어주는 두번째 탐색과정으로 구성하였다.

(a) Effect of Delaunay radius test

(left : input node data, middle : circumcircle grouping, right: final mesh)

(b) Effect of arbitrary triangulation without Delaunay test

Figure 3. Example of Delaunay radius test.

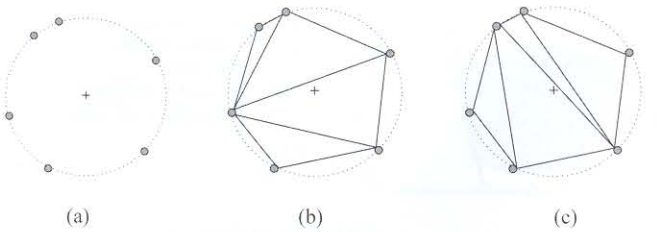


Figure 4. A concentric circle problem.

2.3. 삼각화 과정

삼각화 과정에서 바운더리의 형상을 유지하면서 메쉬를 생성하기 위해서는 선택된 삼각형 집합이 바운더리 내에 존재하는 지를 확인하는 과정이 필요한데 이를 point-in-polygon test로 처리하였다. 주어진 점이 다각형 내에 존재하는 지의 여부는 잘 알려진 ray-casting 알고리즘[8]으로 계산하였다.

전체적인 삼각화 과정은 *Figure* 5에 나타내었다. 주어진 점의 집합에서 임의의 세개의 점 집합 (i, j, k)를 선택(incremental searching)한 뒤, 그림에서와 같은 검사 과정을 통해 새로운 요소가 되는 집합은 기록하고 그렇지 않은 집합은 버리는 과정을 반복함으로써 삼각화가 진행된다. 이러한 경우 점의 수가 N일 때 집합을 구성할 수 있는 전체 경우의 수는 $O(N^3)$이 되어서 매우 비효율적이므로, 구역별 삼각화와 선분기반의 반평면 탐색이라는 두가지 방법으로 가능한 삼각형 조합의 수를 줄였다.

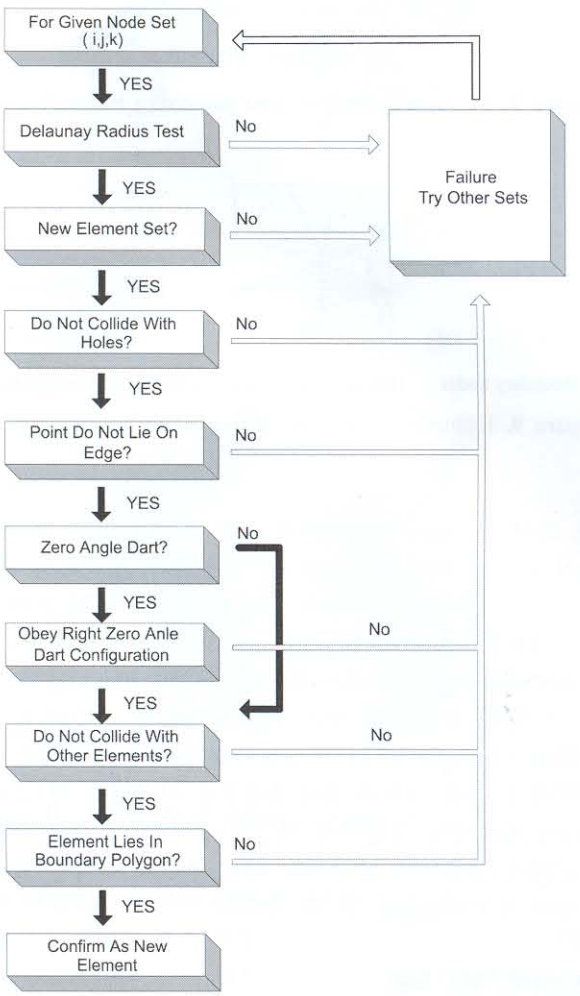


Figure 5. Triangulation procedure.

첫번째로 구역별 삼각화(local triangulation)는 점 집합 (i, j, k)를 탐색함에 있어서 전체 구역에 대해서 탐색하지 않고, 전체 영역을 *Figure* 6과 같이 바둑판 모양으로 일정한 크기의 구역으로 나누어서 탐색하는 방법이다. Delaunay 알고리즘은 특정한 근처에 있는 점들로부터 이등변 삼각형에 가까운 삼각형을 탐색하게 되므로, (i, j, k)로부터 멀리 떨어진 점들은 다음 탐색에 필요가 없으므로 구역별 삼각화를 하게 되면 세부 구역의 수만큼 경우의 수가 줄어 들게 된다. 집합 (i, j, k)를 전체 집합에 대해 탐색하게 되면 경우의 수가 $O(N^3)$이 되지만, 전체 구역을 M개로 나눈 경우 경우의 수는 $O(N^3/M)$이 된다.

두번째로 선분기반의 반평면 탐색(edge-based half-plane searching)은 집합 (i, j, k)로 삼각형 요소를 구성한 뒤 다음 삼각형 집합 (m, n, o)를 탐색할 때 탐색 영역을 이미 구성된 선분 jk의 삼각형 바깥쪽 영역으로 제한하는 방법이다. *Figure* 7(a)에서와 같이 점의 집합이 주어지고 *Figure* 7(b)에서 (i, j, k)의 요소를 구성하였다고 할 때, 다음 요소의 집합은 (j, k, m)로서 경우의 수가 $O(N)$이 되고, 점 m은 삼각형 요소 (i, j, k)의 반대쪽에 있어야 하므로 점 M의 검색은 평균적으로 $O(N/2)$가 된다.

노드의 수에 따른 삼각화에 걸리는 시간을 *Table* 1에 나타내었다. 두 가지 최적화 방법을 이용한 결과 $O(N^3)$이었

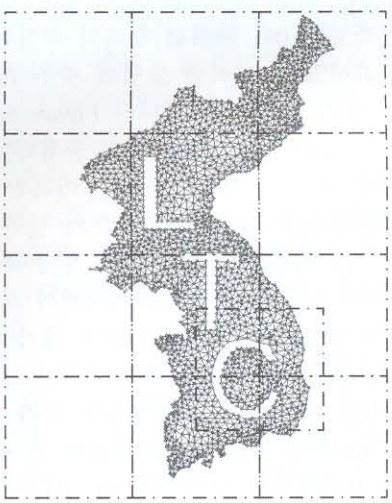


Figure 6. Local triangulation.

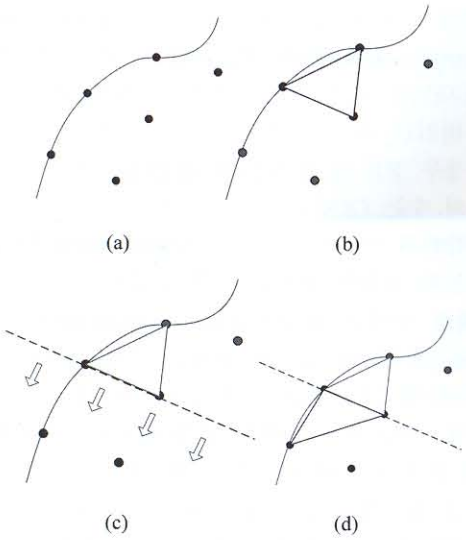


Figure 7. Edge-based half-plane searching.

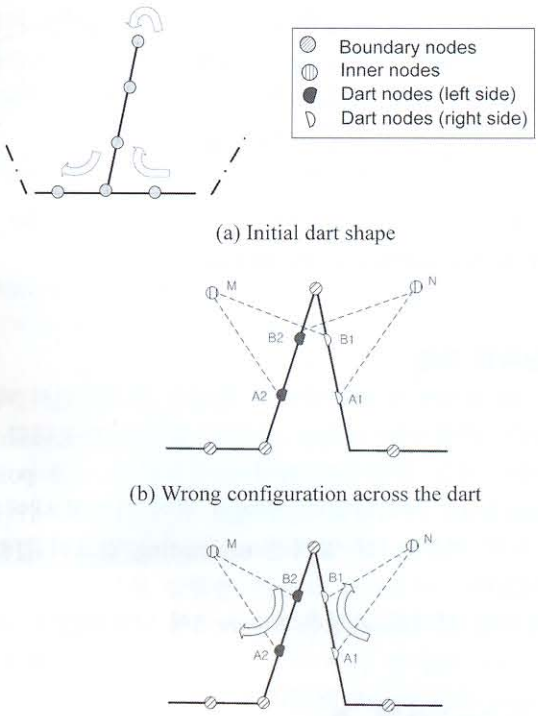
Table 1. Triangulation time

Number of nodes	Number of elements	Time (seconds)
43	43	0.15
119	193	0.94
415	772	1.36
1068	2064	8.57

던 경우의 수를 $O(N^2)$으로 낮춤으로써 복합한 도형의 경우에도 효과적인 삼각화가 가능하였다.

2.4. 내부 홀과 다트를 갖는 패턴의 메쉬 생성

의복 패턴의 가장 큰 특징이 2차원 평면 패턴을 3차원 입체 형상으로 만들기 위해 다트가 주어진다는 점이다. 일반적인 경우 다트의 각은 입체를 형성하기 위해 충분한 정도로 주어지지만, 이상적인 메쉬 알고리즘을 위해서 *Figure* 8(a)와 같이 각도가 0인 가위질 형태의 극심한 정도의 다트까지도 고려하였다. 이러한 도형의 경우 *Figure* 8(a)의 점 A, B는 하나의 점 좌표가 아니라 *Figure* 8(b)에서와 같이 두 개의 점들(A1-A2, B1-B2)끼리 중첩되어 있음을 알 수있다. 기존의 메쉬 알고리즘에서는 이러한 기하학적인 문제가 발생하지 않으므로 이러한 0도의 날카로운 각도의 다트 형상은 처리하지 못하였으나 본 연구에서는 중첩된 점들로 이루어진 도형에 대해서도 외곽선의 회전방향을 고려하여 삼각화 알고리즘을 개발하였다. 예각 다트 부분의 특징은, 점 A, B가 중첩되어 있기는 하지만 그들이 외곽선 위에 존재하므로, 외곽선의 회전방향을 따라 점을 추적하면 점 A1과 A2 또는 B1과 B2를 구분해 낼 수 있다는 것이다. *Figure* 8(c)에서와 같이, 점 A1, B1은 같은 방향의 외곽선위에 있으므로 점 N에 대해 삼각형 형성이 가능하다. 그러나 *Figure* 8(b)에서와 같이 점 A1, B2는 다른 방향의 외곽선 위에 있으므로 점 N에 대해 삼각화를 할 경우 잘못된 메쉬가 형성된다. 이러한 과정을 통해 얻어지는 0도 각도의 다트의 예를 *Figure* 9에 나타내었다. *Figure* 9(a)는 외곽선을 나타내는 점의 집합인데, 0도 다트의 점들은 두 개씩 중첩되어 있다. *Figure* 9(b)는 삼각화가 끝난 상태를 나타내는데, 다트의 삼각화가 제대로 되었는지를 확인하기 위해 다트에 좌우방향으로 인장력을 한 그림이 *Figure* 9(c)이다. 그림에서 확인할 수 있듯이 점의 좌표가 중첩되어 있음에도 불구하고 다트 부근의 삼각형 요소가 정확히 형성되었음을 알 수 있다.

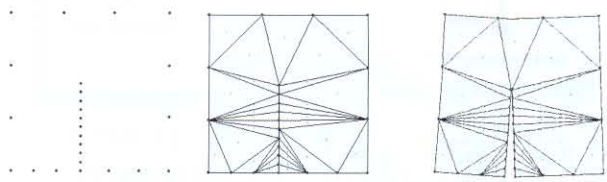


(a) Initial dart shape

(b) Wrong configuration across the dart

(c) Right configuration

Figure 8. Schematic view of zero angle dart triangulation.



(a)Boundary nodes (b)Constructed mesh (c)Laterally stretched pattern

Figure 9. Example of meshing pattern with zero-angle dart.

2.5. 의복 패턴 적용

고안된 알고리즘이 실제 의복 패턴에 있어서도 적용되는

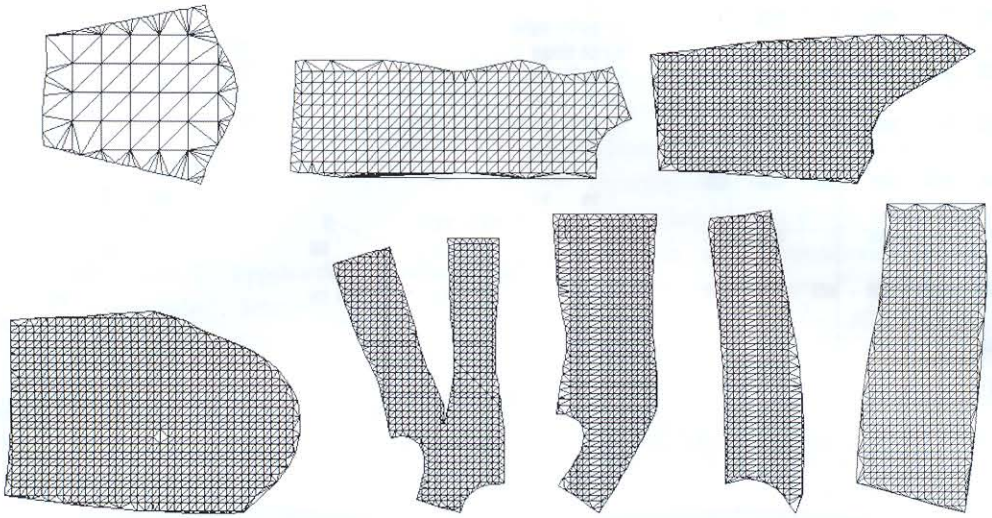


Figure 10. Mesh examples of patterns designed by commercial tools.

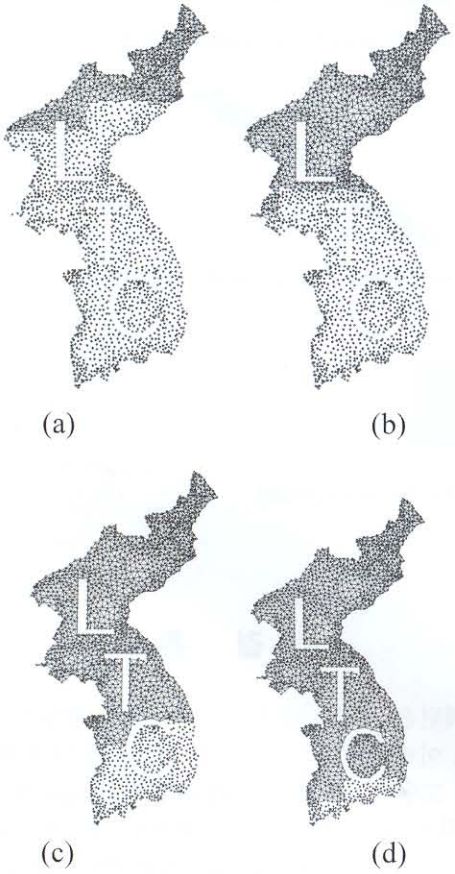
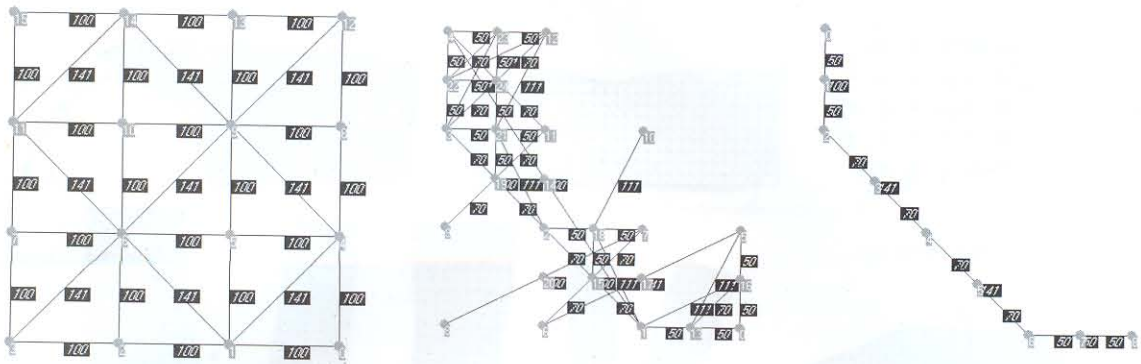


Figure 11. Sub-regional Delaunay triangulation.

지를 시험하기 위하여 Gerber, PAD, Yuka 등의 상용 시스템에서 디자인한 패턴 형상들을 가지고 삼각화를 시행하였다. *Figure* 10과 같이 다트가 없는 패턴의 경우에도 삼각형 요소가 잘 형성됨을 알 수 있었다. *Figure* 11은 국소적 삼각화를 이용할 때 복잡한 도형의 메쉬 생성 과정을 보여주고 있다.

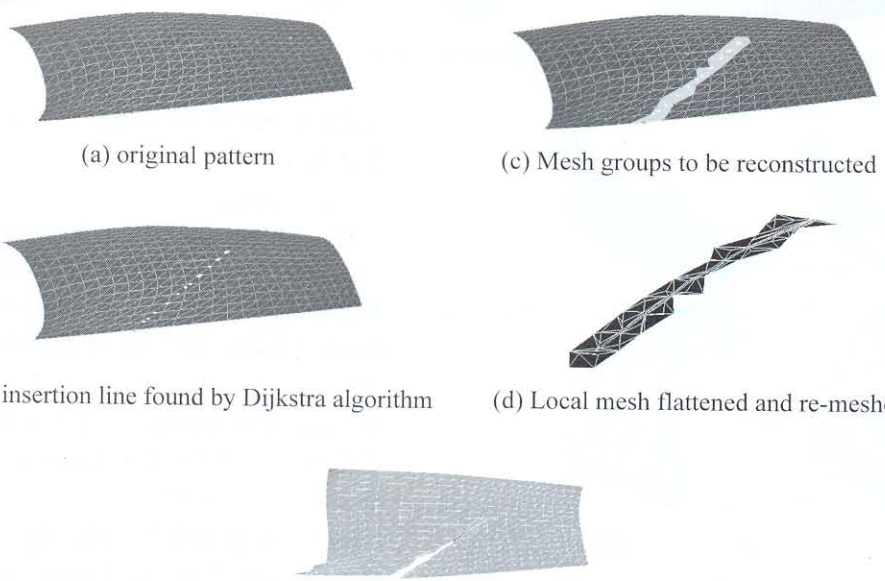
2.6. Adding Dart Line Interactively

0도 다트가 있는 패턴의 삼각화를 테스트하기 위해 다트 라인을 패턴 디자인이 끝난 후에 동적으로 형성하였다. *Figure* 12(a)와 같이 3차원 공간에 미리 형성된 패턴 삼각 메쉬를 위치시키고, 자르고자 하는 다트의 라인의 끝점을 마우스로 선택한 다음 두 점간의 최단경로를 Dijkstra 알고리즘[9]으로 탐색하였다. Dijkstra 알고리즘은 그래프 상의 두 노드 간의 최단 경로를 구하는 알고리즘으로서, *Figure* 12(a)에서와 같이 초기 그래프의 선분에 각 노드간 거리(또는 가중치)가 주어지면 *Figure* 12(b)와 같이 첫번째 탐색 그래프에서 최단경로 주변의 선분만을 탐색하고, *Figure* 12(c)의 두번째 탐색에서 최종적인 최단경로를 찾는 방법이다. 마찬가지 방법으로 패턴 상의 다트 라인을 탐색한뒤 (*Figure* 13a-c), 다트 라인을 포함하는 삼각형 요소들의 부분집합을 2차원 xy 평면 위로 투영시켰다(*Figure* 13d). 이는 3차원 곡면상에서의 메쉬는 곡률을 가지므로 삼각화가 힘들기 때문이다. 2차원 평면상에 투영된 삼각메쉬에서 요소 정보를 삭제하고 탐색된 다트라인의 점들을 삽입하여 다시 삼각화를 시행하였다. 그런 다음 재형성된 다트 주변 메쉬를 원래 패턴 메쉬에 삽입하였다. 최종적으로 생성된 패턴의 삼각화를 확인하기 위해서, 패턴을 다트를 중심으로 좌우로 인장력을 가하여 펼친 것이 *Figure* 13(e)이다. 따라서 본 연구에서 고안한 메쉬 알고리즘이 0도 각도의 예각을 갖는 다트라인의 삼각화도 잘 처리할 수 있음을 알 수 있었다. *Figure* 14는 메쉬화된 패턴들을 삼차원 상에서 배치하여(a) 봉제조건을 부여하였을 때(b)의 형상을 나타내고 있다.



(a) Initial mesh data (0th graph) (b) 1st graph (c) 2nd graph

Figure 12. Example of Dijkstra algorithm until 2nd loop.

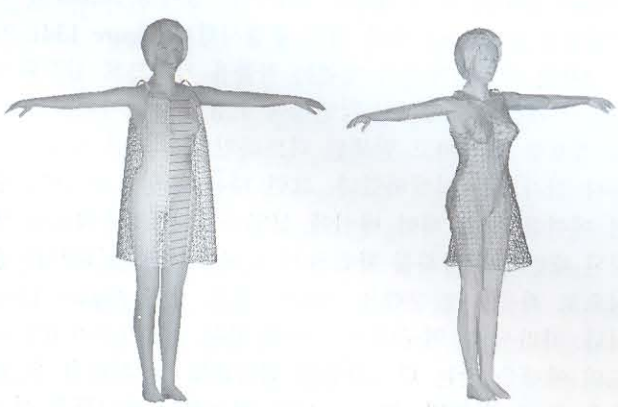


(a) original pattern (c) Mesh groups to be reconstructed

(b) dart insertion line found by Dijkstra algorithm (d) Local mesh flattened and re-meshed

(e) Laterally stretched pattern to verify the inserted dart

Figure 13. Example of 3D dart insertion and lateral stretching.



(a) Intial alignment (b) After sewing

Figure 14. Example of 3D pattern alignment and sewing.

3. 결 론

의복 패턴은 홀과 다트와 같은 특이한 형상을 포함하고 있으므로 이에 적합한 메쉬 생성 알고리즘이 필요하다. 본 연구에서 고안한 삼각화 알고리즘은 Delaunay 삼각화를 이용하여 최소 0도의 각을 갖는 예각의 다트까지도 처리할 수 있으며 다트를 포함하지 않는 일반적인 의복 패턴의 경우도 메쉬를 성공적으로 생성하였다. incremental searching을 이용한 삼각화법은 경우의 수가 많아서 연산량이 많은 단점이 있으므로 국소적 삼각화와 반평면삼각화법을 이용하여 경우의 수를 노드 수의 제곱 이하로 낮추었다. 본 알고리즘을 3차원 의복 패턴의 동적인 다트 형성에 응용한 결과 3차원 의복 패턴 디자인에 있어서도 효과적으로 응용

될 수 있을 것으로 생각된다.

감사의 글: 본 연구는 과학기술부/한국과학재단 우수연구센터육성사업의 지원으로 수행되었음(R11-2005-065).

참고문헌

1. M. Bern, D. Dobkiny, and D. Eppstein, "Triangulating Polygons without Large Angles", International Journal of Computational Geometry and Applications, 1995, 5, 171-192.
2. J. R. Shewchuk, "Delaunay Refinement Algorithms for Triangular Mesh Generation", Computational Geometry: Theory and Applications, 2002, 22(1-3), 21-74.
3. D. E. Breen, D. H. House, and M. J. Wozny, "A Particle-Based Model for Simulating the Draping Behavior of Woven Cloth", Text Res J, 1994, 64(11), 663-673.
4. R. A. Dwyer, "A Faster Divide-and-conquer Algorithm for Constructing Delaunay Triangulations", Algorithmica, 1987, 2, 137-151.
5. S. Fortune, "A Sweepline Algorithm for Voronoi Diagrams", Algorithmica, 1987, 2, 153-174.
6. K. L. Clarkson and P. W. Shor, "Applications of Random Sampling in Computational Geometry(II)-Discrete and Computational Geometry", 1989, 4, 387-421.
7. P. Su and R. L. Scot Drysdale, "A Comparison of Sequential Delaunay Triangulation Algorithms", Computational Geometry : Theory and Applications, 1996, 7(5/6), 361-386.
8. J. O'Rourke, "Computational Geometry in C", Cambridge University Press, 1994, pp.52-69.
9. E. W. Dijkstra, "A Note on Two Problems in Connection with Graphs", Numeriche Math, 1959, 1, 269-271.