

트림된 NURBS 곡면의 효율적인 삼각화 알고리즘

정재호*, 박준영**

An Efficient Triangulation Algorithm for Trimmed NURBS Surfaces

Jung, J. H* and Park, J. Y.**

ABSTRACT

We propose an algorithm for obtaining a triangular approximation of a trimmed NURBS surface. Triangular approximation is used in the pre-processing step of many applications such as RP(Rapid Prototyping), NC(Numerical Control) and FEA(Finite Element Analysis), etc. The algorithm minimizes the number of triangular elements within tolerance and generates a valid triangular mesh for STL file and NC tool path generation. In the algorithm, a subdivision method is used. Since a patch is a basic element of triangular mesh creation, boundary curves of a patch are divided into line segments and the division of curves is applied for the interior of the surface. That is, boundary curves are subdivided into line segments and two end points of each line segment are propagated to the interior of the surface. For the case of a trimmed surface, triangulation is carried out using a model space information. The algorithm is superior because the number of elements can be controlled as the curvature of the surface varies and it generates the triangular mesh in a trimmed region efficiently. To verify the efficiency, the algorithm was implemented and tested for several 3D objects bounded by NURBS surfaces.

Key words : Triangulation, NURBS, Trimmed surfaces, Mesh, Rapid prototyping, NC

1. 서론

삼각화는 유한 요소 해석, 수학적 분석, 컴퓨터 그래픽스 분야의 가시화, CAGD (Computer Aided Geometric Design), Offset곡면의 생성, NC 공구경로의 생성, RP에서 사용되는 STL파일 등에 적용된다^[7,8]. 이와 같이, 삼각화는 사용되는 분야가 다양하며 그에 따라서 수행방법도 달라지게 된다. 예를 들어, 유한 요소 해석이나 수학적 분석 등에서는 정삼각형에 가까운 요소를 생성하는 것이 중요한 의미가 있을 수 있으며, 그러한 요소를 생성하기 위해서 주로 Delaunay 삼각화가 널리 사용되고 있다^[7]. NC 공구경로의 생성이나 RP의 STL파일 등에 사용되기 위해서는 허용 공차내에 들면서 삼각형의 개수를 최소화 하는 것과 메쉬의 모양을 균일하게 하는 것도 고려 되어야 할 사항들이다. 왜냐하면 메쉬의 모양

이 균일하지 않으면 공구경로나 STL파일에서 오차가 불균일하게 발생할 수 있기 때문이다. 즉, 불균일한 메쉬를 이용할 경우에는 가공된 형상의 표면이 불규칙하게 생성될 수 있다. 균일한 메쉬를 생성하기 위해서는 격자이용법과 같은 방법들이 이용되고 있다^[10].

일반적으로 복잡한 형태의 구조물을 모델링할 때 전체 영역을 많은 수의 영역들 즉, 패치들로 나눈 후 각 패치상에 삼각요소나 사각요소를 생성하게 된다. 이러한 패치들은 보통 세 개 또는 네 개의 경계 곡선으로 형성되는 곡면이며 삼각화의 기본 단위로 사용된다^[11]. 본 논문에서 제시된 알고리즘도 이러한 패치를 기본으로 한다. 그리고 NC공구경로 생성이나 STL파일에서 사용하기 위해서 삼각형의 개수를 허용 공차내에서 최소한 줄이며 균일한 삼각메쉬를 생성하는 것을 목표로 하고 있다.

지금까지의 선행연구들에서 3차원 곡면상에서의 삼각화는 주로 매개공간상에서 이루어지거나 모델공간에서 이루어지는 두 가지의 방법이 주로 사용되었

*학생회원, 동국대학교 산업공학과

**종신회원, 동국대학교 산업공학과

다. 매개공간 상에서의 삼각화는 빠른시간내에 이루어 질 수 있는 장점이 있는 반면에 메쉬의 모양을 조절하기 어려운 단점을 지니고 있다. 모델공간 상에서의 삼각화는 메쉬의 모양을 조절하기 용이한 한편 시간이 오래 걸리며 매개변수식에 의한 표현방식을 유지하기 어려운 단점이 있다. 지금까지의 선행 연구들에서는 트림 곡면의 삼각화가 주로 매개 공간 상에서 수행되었기 때문에 삼각메쉬의 모양을 조절하거나 곡면의 곡률에 따른 요소의 개수를 변화시키는 것이 매우 어려웠다. 하지만 본 논문에서 제안된 알고리즘은 곡률에 따라 삼각요소의 개수를 조절함으로써 데이터의 양을 가능한 최소로 하며 곡률의 변화가 없는 영역에서는 균일한 메쉬를 효율적으로 생성한다.

2. 기존 연구

2차원 평면상에서 주어진 점들로부터 삼각 메쉬를 생성하는 경우에, Delaunay 삼각화는 정삼각형에 근접한 모양의 요소들을 생성하는(aspect ratio가 큰) 특징이 있다. 따라서 정삼각형의 형상을 요구하는 응용 분야에 Delaunay 삼각화가 널리 사용되며, 구면 방법으로는 일반적으로 Voronoi polygon과 Watson의 알고리즘이 가장 많이 이용된다. Voronoi polygon은 각 점들을 잇는 선분의 수직 이등분선에 의해서 만들어지며 Voronoi polygon의 한 꼭지점을 공유하고 있는 서로 인접한 세 개의 다각형 내부에 존재하는 점을 연결하여 삼각요소를 생성한다. Watson의 알고리즘은 서로 다른 세 점으로 구성된 삼각요소에서 그 외접원 내에 다른 점이 포함되지 않도록 하는 것이다^[10]. 이와 같이 메쉬를 생성하면 곡률반경을 고려하여 삼각요소를 생성하기가 어렵게 된다^[7].

트림 곡면의 경우, 시스템간의 데이터 교환시에는 주로 모델공간상의 정보들만 교환되기 때문에 트리밍 곡선과 트림곡면에 대한 매개 변수식의 표현이 필요하게 된다. 이 과정은 많은 계산량이 요구되어 여러 가지 기법이 개발되었다^[1-2]. 최근에는 해석을 목적으로 한 트림곡면상에서 사각형 메쉬를 생성하는 기법이 소개되기도 하였다^[2]. 그 이외의 방법으로, 트리밍 곡선을 모델공간상의 곡선이 아닌 매개공간상의 곡선으로 정의하는 것이 있다^[5-6,8]. 이 경우에는 모델 공간상의 정보만이 입력데이터로 주어지면 트리밍 곡선의 좌표를 곡면에 대한 매개 좌표로 변환하는 과정이 요구되어진다. 본 논문에서는 매개 공간상에서 삼각화를 수행하여 모델 공간상으로 매핑

하는 기법을 사용하고 있으며, 모델공간상의 정보를 이용하여 곡면의 곡률 변화에 따라서 메쉬의 생성 정도를 조절할 수 있게 한다. 트리밍 곡선과 트림곡면의 교차를 계산하는 과정에서는 속도가 빠른 새로운 기법을 제시하고 있다.

3. NURBS 곡면에서의 삼각 메쉬 생성

NURBS(Non-Uniform Rational B-Spline)는 충분한 연속성 보장, 국부적인 변형 가능, 곡면식의 일반성, 곡면제어의 편이성 등의 여러 가지 장점을 가지고 설계를 보다 용이하게 하므로 곡면설계에 자주 이용된다^[3-4]. NURBS곡면과 곡선의 일반식은 식 (1), (2)와 같이 매개 변수 u, v 에 의해 표현된다^[3-4].

$$C(u) = \frac{\sum_{i=0}^n w_i P_i N_{i,p}(u)}{\sum_{i=0}^n w_i N_{i,p}(u)} \quad (1)$$

$$S(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^n w_{i,j} P_{i,j} N_{i,p}(u) N_{j,q}(v)}{\sum_{i=0}^n \sum_{j=0}^n w_{i,j} N_{i,p}(u) N_{j,q}(v)} \quad (2)$$

위 식들에서,

$C(u)$: NURBS 곡선식

$S(u, v)$: NURBS 곡면식

p, q : 차수(degree)

$P_i, P_{i,j}$: 조정점(control points)

$w_i, w_{i,j}$: 가중치(weights)

$N_{i,p}(u), N_{j,q}(v)$: B-스플라인 기저 함수

본 연구에서는 NURBS의 매개 변수를 이용하여 u, v 매개 공간상에서 삼각요소를 생성한 후 모델 공간상으로 매핑(mapping)한다. 즉, Fig. 1과 같이 u, v 평면의 각 경계점들($u=0, 1$ 과 $v=0, 1$)을 포함한 모든 점들이 모델 공간상의 곡면의 점들과 대응되기 때문에 모델 공간상의 곡면에 대한 정보를 입력받아서 매개 공간상의 평면으로 변환한 후에 메쉬를 생성하게 된다. 이러한 삼각요소의 생성과정을 간단히 설명하면 다음과 같다. 우선 등매개 곡선을 이용하면 삼각요소의 크기나 개수 등을 쉽게 조절할 수 있기 때문에 그것을 이용하여 곡면을 사각형으로 분할한다. 이렇게 분할이 이루어지면 사각형을 다시 각각 2개로 분할하여 삼각요소를 생성하게 된다. 마지막으로 이것을 모델공간상의 좌표로 변환하면 삼각화가 완료된다. 이 과정에서 곡면이나 곡선들의 조

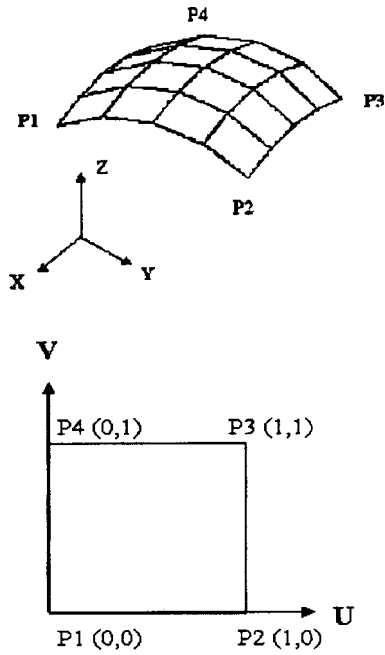


Fig. 1. Surface patch in a model and parametric space.

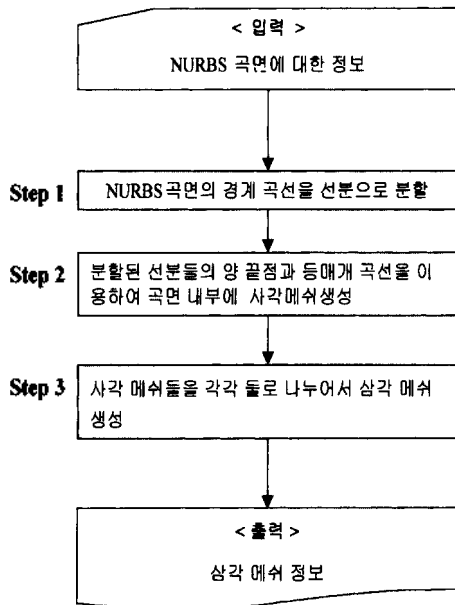


Fig. 2. Triangulation algorithm on a 4-sided surface patch.

정점들은 상용 CAD 시스템을 이용하여 얻었으며, 주어진 매개 변수값을 이용하여 곡면상의 좌표를 구하는 과정에서는 Nlib¹⁾가 사용되었다.

¹⁾NlibNURBS geometry library(Geomware사): NURBS곡면이나 곡선 등의 표현, 조정, 처리 등의 작업을 위한 library.

3.1 사각 곡면 패치에서의 삼각 메쉬 생성

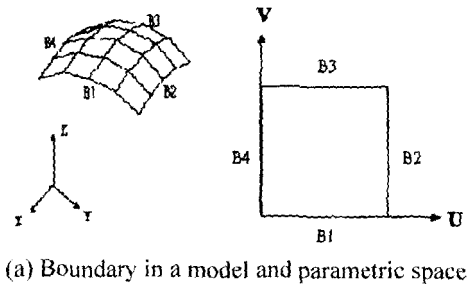
사각 곡면 패치는 4개의 경계 곡선으로 이루어지며, 매개공간 상에서 정사각형을 이루는 곡면 패치를 의미한다. 이 절에서는 사각 곡면 패치에 적용되는 알고리즘이 설명될 것이며 트림 곡면에 적용되는 알고리즘은 다음 절에서 다루어지게 된다. 이러한 사각 곡면 패치의 삼각메쉬 생성 알고리즘은 Fig. 2와 같이 설명될 수 있다.

3.1.1 경계곡선을 선분으로 근사

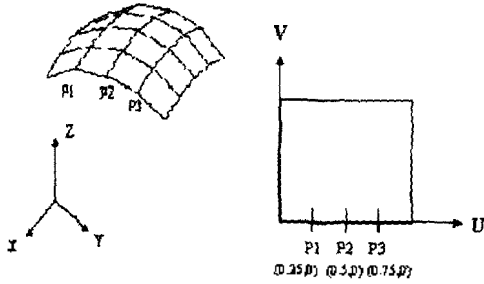
본 연구에서 제안하는 메쉬 생성 알고리즘은 크게 외부의 경계 곡선을 선분으로 근사하는 과정과 곡면 패치 내부에 존재하는 등매개 곡선을 선분으로 근사하는 과정으로 나눌 수 있다. 본 절에서는 우선 경계 곡선을 선분으로 근사하는 과정을 설명하기로 하자.

패치(patch)의 경계곡선은 Fig. 3(a)에서 처럼 매개공간의 u, v 평면에서 경계선분으로 대응된다. 하나의 경계곡선에서 곡선의 양 끝점을 잇는 선분을 구하고, 경계선분이 주로 3차식으로 정의되는 점을 고려하여 매개공간상의 경계선분에서 1/4, 2/4, 3/4 위치에 있는 3점들에 대한 모델 공간상의 좌표를 구한다(Fig. 3(b)). 이와같이 구한 선분과 3점의 좌표를 이용하여 모델 공간상의 경계 곡선과 선분이 이루는 수직 거리를 구한다(Fig. 3(c)). 이때 선분의 가운데 점에서만 곡면과의 거리를 구하면 Fig. 4와 같이 근사가 정확하게 이루어지지 않는 경우가 존재하게 된다. 즉, 경계곡선상의 $u=1/2$ 인 점에서 곡면과의 수직 거리는 공차범위 이내에 존재하므로 더 이상 경계곡선의 분할이 이루어지지 않게 된다. 이러한 경우에도 보다 정확한 경계곡선을 표현하기 위해서 $u=1/4, 3/4$ 인 점을 선택하여 각 점으로부터 곡면과의 수직 거리를 계산한다. 계산된 값이 주어진 공차보다 적으면 더 이상 작은 선분들로 나누지 않고, 그와 반대로 수직 거리가 공차보다 클 때는, 근사화된 경계곡선이 공차내에 들게 하기 위해서 고려하고 있는 매개변수 구간의 중간위치에 한 점을 삽입하며(Fig. 3(d)), 이와같은 과정을 공차를 만족할 때까지 반복한다.

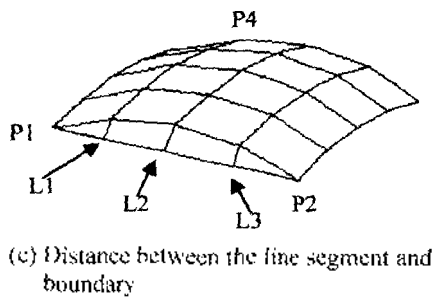
이상과 같이 곡면의 경계곡선들은 Fig. 3(e)에서 처럼 선분들로 모두 분할된다. 이 때, 마주 보는 선분들에서 점의 개수가 다르면 등매개 곡선을 생성할 수 없으며 사각 메쉬의 모양이 균일하지 않게 된다. 결과적으로 사각메쉬를 분할하여 생성되는 삼각메쉬의 모양도 균일하지 않게 되기 때문에 점의 위치와 개수의 조정이 필요하게 된다. Fig. 5는 Fig. 3(e)에서 마주보는 경계곡선상의 점의 개수를 같게 하기 위하여 매개곡선 상에서 $u=1/2$ 과 $u=3/4$ 의 위치에 2



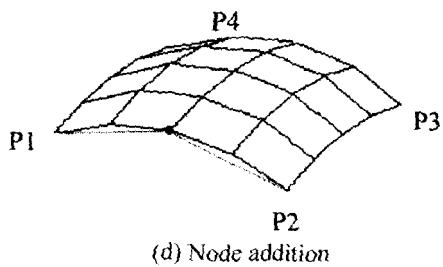
(a) Boundary in a model and parametric space



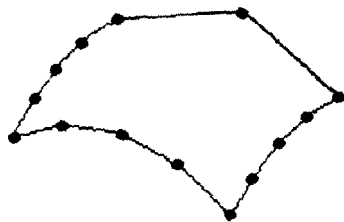
(b) 3-point in a model and parametric space



(c) Distance between the line segment and boundary



(d) Node addition



(e) Result of the approximation

Fig. 3. Approximation of the boundary curve

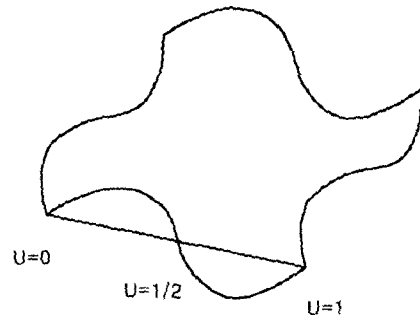


Fig. 4. Distance between the line segment and boundary curve.

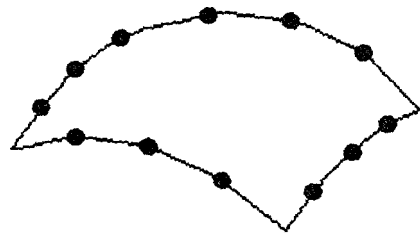


Fig. 5. Boundary with node modification.

점을 추가하여 조정한 결과를 보여주고 있다.

지금까지는 경계 곡선에 대한 근사만을 고려했기 때문에 경계 곡선과 곡면 내부의 곡률이 크게 다를 경우에는 별도의 과정이 필요하다. 이 과정을 Fig. 6과 같이 경계 곡선은 곡률이 매우 작지만 곡면의 내부는 곡률이 큰 경우의 예를 들어 설명하기로 한다. 이 예에서 경계 곡선을 직접적으로 분할하는 것은 필요하지 않지만 곡면의 내부에 존재하는 곡선들에 대해서는 선분으로 분할하는 것이 필요하게 된다. 그 절차는 다음과 같다. 우선, 앞서 언급한 방법에 의하여 경계 곡선에 새로운 노드의 삽입이 필요한가를 검사해야 한다. 새로운 노드의 삽입이 필요한 경우는 Fig. 6과 같이 매개 공간상에서 1/4, 1/2, 3/4의 위치에 놓인 곡면 내부의 곡선들에 대해서도 앞에서

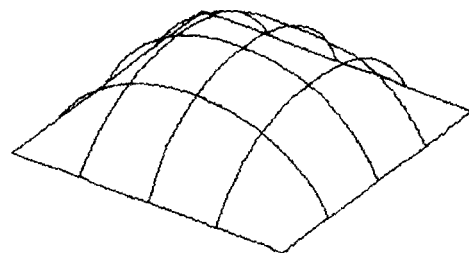


Fig. 6. Surface with large curve difference.

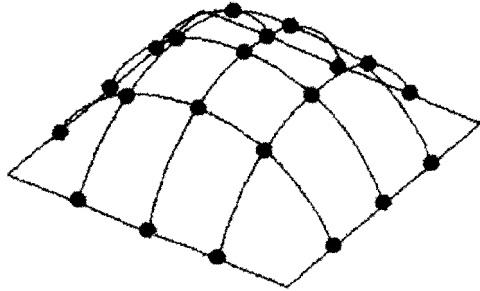


Fig. 7. Modification of nodes number and position.

설명한 알고리즘을 동일하게 적용시킨다. 즉, Fig. 6과 같이 경계 곡선과 곡면 내부의 곡물의 차이가 큰 곡면은 경계 곡선(4개)과 곡면 내부의 곡선(6개)을 모두 선분으로 분할하는 과정을 거쳐야 한다.

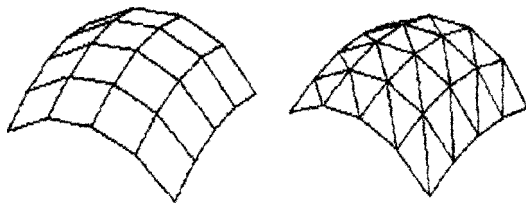
위의 과정에 따라서 Fig. 6과 같은 곡면에 대한 근사화가 이루어지게 된다. 이 경우에서도 앞서 설명한 것과 동일한 이유로 점의 위치와 개수를 조정해야 한다. Fig. 7은 Fig. 6의 곡면에서 점들의 위치와 개수를 조정된 결과를 보여주는 것이다.

3.1.2 삼각매쉬의 생성

앞 절의 절차에 따라 곡면의 경계 곡선(또는 내부의 곡선)이 선분들로 모두 분할되고 이것을 이용하여 매쉬를 생성하게 된다. 이 과정은 다음과 같다. 분할된 선분들의 양 끝점을 지나는 등매개 곡선을 곡면 내부에 적용한다. 즉, 마주 보는 점들을 연결해서 등매개 곡선을 만들어 주면 사각요소를 생성할 수 있으며(Fig. 8(a)). 이러한 사각요소들을 각각 둘로 나누어주면 삼각요소를 생성할 수 있다(Fig. 8(b)).

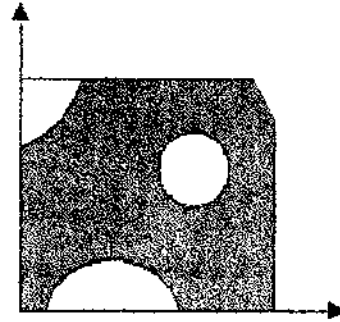
3.2 트림 곡면에서의 삼각 매쉬 생성

트림 곡면이란 곡면상의 곡선에 의해 곡면의 내부 또는 경계에 구멍이 뚫려 있거나 잘려진 곡면을 의미한다. Fig. 9는 이러한 트림 곡면의 예를 보여주고 있는 것이다. Fig. 9(a)는 매개 공간상에 존재하는 트

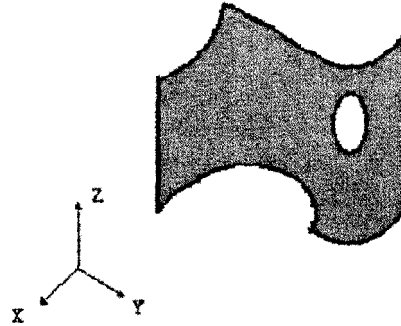


(a) Quadrilateral mesh (b) Triangular mesh

Fig. 8. Triangular mesh generation.



(a) Trimmed surface in a parametric space



(b) Trimmed surface in a model space

Fig. 9. Example of a trimmed surface.

림 곡면이며, 이를 3차원 상의 좌표값으로 변환하면 Fig. 9(b)와 같은 결과를 얻을 수 있다.

일반적으로 트림 곡면과 트리밍 곡선은 다른 매개 공간상에서 정의된다. 따라서 선행연구들에서는 여러 가지 방법을 이용하여 동일한 매개 공간상에서 표현 되도록 하였다. 하지만 앞절(2절)에서 설명한 바와 같이, 이 과정은 시간이 많이 소요된다는 단점이 있기 때문에 효율적이지 않다. 따라서 본 연구에서는 이러한 과정 없이 3차원 상의 점을 이용하여 트림 곡면에 대한 삼각 매쉬를 생성하는 효율적인 알고리즘을 제시하고 있다.

3.2.1 교차영역 판별

트림 곡면에서 트리밍 곡선의 외부에 해당되는 영역은 곡면이 존재하며 삼각요소를 생성해야 하는 부분이다. 이 영역을 판별하기 위해서는 사각 곡면 패치상에 놓인 트리밍 곡선의 위치를 알아야 하는데, 이를 위해서 곡선과 곡면의 교차영역을 구하는 과정이 필요하다. 그리고 교차영역에서 삼각매쉬를 생성하기 위해서는 트리밍 곡선의 외부에 있는 점들을 판별해야 한다. 이러한 트림 곡면의 삼각화 과정을 설명하기 위해서 Fig. 10의 곡면

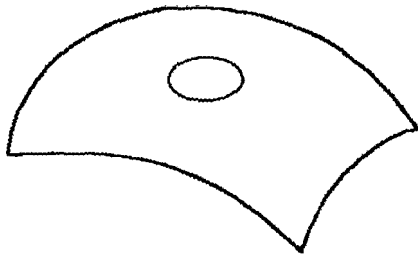
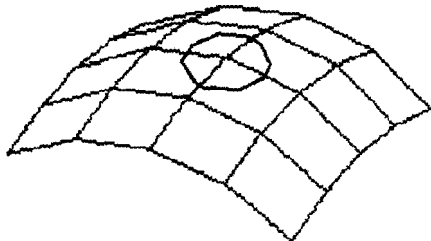


Fig. 10. Trimmed surface with a hole.

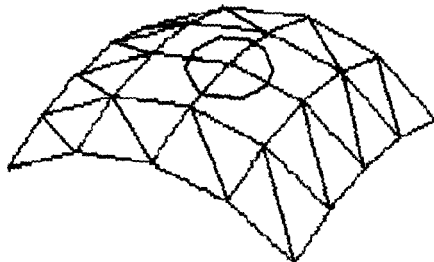
우선, 트리밍곡선을 선분들로 나누고 곡면은 사각 메쉬들로 분할한다. 이 단계까지는 앞절에서 설명한 사각 곡면 패치에 대한 방법과 동일하며 Fig. 11(a)와 같은 결과를 얻을 수 있다. 곡면과 곡선간의 교차를 직접 검사하는 것은 그들의 차수가 높아지면 시간과 노력이 많이 요구되기 때문에 비효율적이다. 하지만 모든 곡면과 곡선들이 선분들로 분할되면 선분들간의 교차를 검사하는 것이 되기 때문에 상대적으로 교차 검사가 간단해진다. 이와 같이 곡면에서 트리밍 곡선과 교차하는 사각형들을 찾은 후에 교차하지 않는 사각 요소들을 각각 들로 나누어 삼각 요소 생성한다. 이 과정에 대한 결과는 Fig. 11(b)에 나타나 있다.

3.2.2 외부점 추출

앞 절에서는 트리밍 곡선과 교차하지 않는 영역에



(a) Approximation of a surface with a trimming curve



(b) Triangulation of no trimming curve region

Fig. 11. Intersection identification and the triangulation.

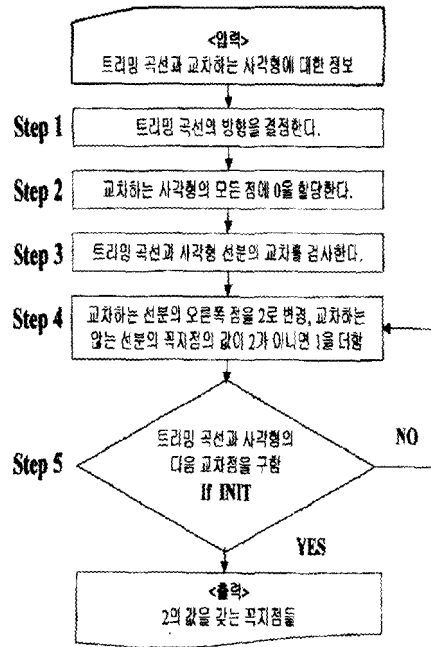


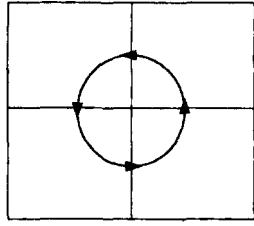
Fig. 12. Algorithm for finding exterior points of a trimming curve.

대해서 삼각 요소를 생성했으며, 본 절에서는 곡선이 곡면과 교차하는 영역에 대한 삼각화를 설명하기로 한다. 삼각 메쉬의 생성이 완료되었을 때, vertex 끼리만 서로 연결되어야 하는 vertex to vertex 조건이 만족되지 않으면 공구 경로에서 중요한 법선 벡터(normal vector)의 방향이 반대로 바뀌는 문제가 발생할 수 있다. 그러한 문제가 생기지 않도록 하기 위해서 사각형의 꼭지점들을 이용하여 메쉬를 생성한다. 메쉬를 생성할 때는 트리밍 곡선의 외부에 있는 점들만이 이용되기 때문에 정확하게 외부점을 찾는 것이 중요하다. Fig. 12는 그 절차를 도식적으로 보여주고 있다.

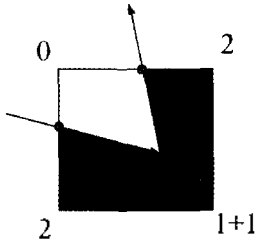
트리밍 곡선과 교차하는 곡면상의 사각요소 중에서 곡선의 외부에 존재하는 점들을 추출하는 알고리즘은 다음과 같다.

Step 1: 트리밍 곡선의 방향을 CCW(시계반대방향)으로 정함, 이때, 곡선의 우측이 곡면의 내부, 좌측이 곡면의 외부가 됨. Fig. 13(a)참조.

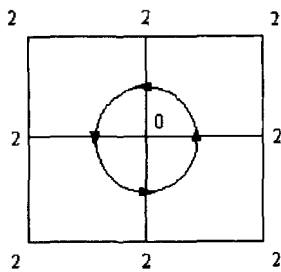
Step 2: 곡면의 경계곡선과 내부의 등매개 곡선을 직선으로 근사화하여 곡면을 사각망으로 형성함(3.3.1과 3.3.2절의 방법을 이용). 트리밍 곡선을 선분들로 근사화하여 사각망과의 교점을 구하고 그 교점들을 포함하는 사각형들을 추출함. 추출된 사각형들의 각



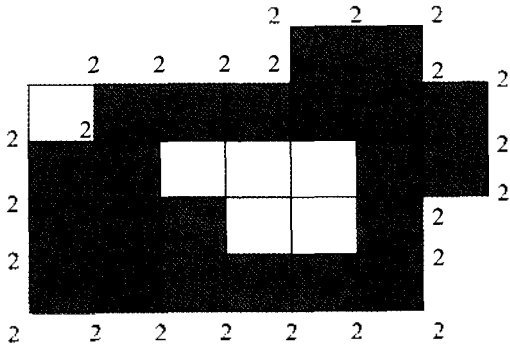
(a) Orientation of a trimming curve



(b) Identification of an exterior node



(c) Result of the algorithm application on a trimmed surface



(d) Intersection region and the node-value

Fig. 13. Procedure for detecting exterior nodes of the trimming curve.

꼭지점에 0을 할당.

Step 3: 트리밍 곡선의 선분과 사각형의 교차점을 검사하여 초기 교차점을 INIT라는 flag를 할당함.

Step 4: 트리밍 곡선 방향을 기준으로, 교차하는

사각형 선분의 우측점에 대한 값을 2로 변경하고, 사각형 내에서 트리밍 곡선과 교차하지 않는 선분의 꼭지점의 값이 2가 아니면 1을 더함.

Step 5: 트리밍 곡선의 방향에 따라 곡선과 사각형의 다음 교차점을 구함. 교차점에 INIT라는 flag가 있으면 꼭지점들의 값이 2인 점들을 출력함. 그렇지 않으면 Step 4를 수행함.

Fig. 13(b)를 이용해서 위의 알고리즘을 적용하면 다음과 같다. 트리밍 곡선과 교차하는 선분의 오른쪽 점인 A에 2의 값을 할당한다. 그리고 교차하는 선분의 맞은 편에 있는 선분 BC의 오른쪽 점인 B에 1을 할당한다. 다음 단계에서는 사각형의 윗쪽 선분과 트리밍 곡선과의 교차가 발생하기 때문에 그 오른쪽 점인 C에 2를, 그 맞은편 선분 AB의 오른쪽 점인 B에 1을 할당한다.

Fig. 11의 곡면에 위 알고리즘을 적용한 결과를 Fig. 13(c)에서 보여주고 있다. 또, 트리밍 곡선이 원이 아닌 자유 곡선일 때 알고리즘을 적용한 예를 Fig. 13(d)에서 보여주고 있다. 트리밍 곡선이 loop를 형성하지 않는 경우에도 위의 알고리즘이 적용될 수 있으며 이에 대한 결과는 다음절에서 보여진다. 이상과 같은 방법으로 알고리즘을 적용시킨 후에 값이 2인 점들만을 이용하여 메쉬를 생성하게 된다.

3.2.3 삼각메쉬 생성

위에 설명한 과정들의 결과로 얻어지는 2의 값을 갖는 점들과 트리밍 곡선상의 점들을 연결하면 트리밍 곡선 주변의 삼각메쉬가 완성된다. 교차 영역에서 메쉬를 생성할 때 사각형의 점들을 이용하기 때문에 Fig. 14에서와 같이 교차영역과 교차하지 않는 영역이 연결되는 부분에서 vertex to vertex조건을 만족시킬 수 있게 된다.

Fig. 15은 Fig. 11을 표시하기 위해서 사용된 NURBS 곡면을 보다 적은 공차로서 삼각메쉬를 생성한 예를

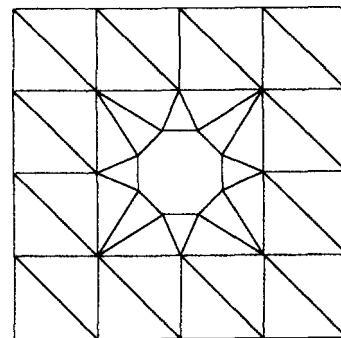


Fig. 14. Triangular mesh generation on a trimmed surface.

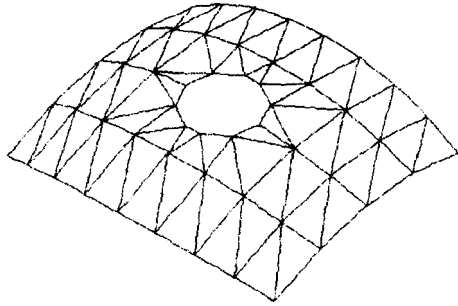


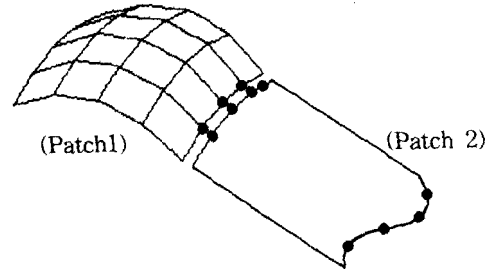
Fig. 15. Mesh with a tight tolerance.

보여주는 것이다. 이와 같이 공차를 조정하면 다양한 크기의 삼각메쉬를 생성할 수 있다.

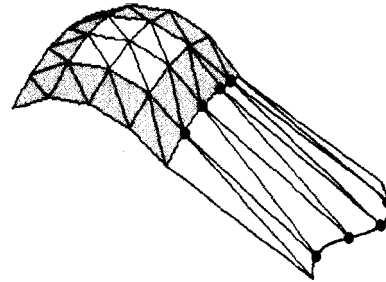
3.3 다수의 곡면패치를로 구성된 곡면상에서의 삼각 메쉬 생성

여러 개의 곡면패치들이 연결되었을 경우 각각의 곡면패치에 대한 삼각 메쉬 생성은 앞서 제안된 방법과 동일한 알고리즘이 적용된다. 인접한 곡면의 경계곡선들을 각각 선분으로 분할한 개수가 같을 경우에는 문제없이 vertex to vertex 조건을 만족시킬 수 있다. 하지만 Fig. 16(a)와 같이 인접한 경계곡선들이 선분으로 분할된 개수가 다른 경우에는 그 조건을 만족하지 않기 때문에 점들을 조정하는 작업이 필요하다. 두 곡면 패치를 공유하는 경계곡선은 표현식이 같은 하나의 곡선이기 때문에 선분으로 분할될 때 동일한 수의 점이 발생되지만 각각의 곡면패치에서 등배개곡선을 유지하려는 과정에서 점의 개수가 달라지는 경우가 생기게 된다. 따라서 이러한 점들의 개수를 조정하는 과정이 필요하며 그 과정은 다음과 같다.

Fig. 16(a)처럼 인접한 경계곡선이 선분으로 분할된 점들의 위치와 개수가 다른 경우가 존재한다고 하면, vertex to vertex 조건을 만족시켜 주기 위해서 Fig. 16(a)에서 (곡면 1)의 분할된 점들의 위치와 (곡면 2)의 점들의 위치를 똑같이 만들어 주어야 한다. 그러나 (곡면 1)의 점들과 (곡면 2)의 점들을 내부까지도 모두 조정하면 (곡면 1)에서 생성되는 삼각 메쉬의 개수가 필요 이상으로 늘어나게 되므로 외부의 점들만을 조정한다. 즉, 가장 바깥쪽 사각메쉬들만 그 점들의 위치를 조정하고 내부 사각형들은 기존의 방법을 이용하여 각각 둘로 나누어 삼각 메쉬를 생성한다. 점들의 조정을 수행한 이후에 점들을 선분으로 연결하여 가장 바깥쪽 메쉬를 완성한다(Fig. 16(b)).



(a) Different nodes generation on a neighboring boundary curve



(b) Triangulation with nodes modification

Fig. 16. Triangulation with neighboring patches.

4. 적용 사례 및 비교

4.1 적용 사례

본 논문에서 제시된 알고리즘의 효율성을 입증하기 위하여 다음과 같은 3차원 모델들에 대해서 삼각화를 수행하였다. Fig. 17(a)와 (b)는 각각 1개와 10개의 패치를 입력하여 얻은 결과를 보여주는 것이다. Fig. 17(c)와 (d)는 각각 트리밍 곡선이 1개와 3개일 경우에 삼각화를 수행한 결과를 보여주고 있다.

Fig. 17(a)에서 보는바와 같이 물체의 곡률이 일정한 경우에는 균일한 모양의 메쉬가 생성되었고, Fig. 17(b)에서 처럼 곡률이 변화가 심한 곡면 상에서는 곡률에 따라 메쉬의 크기가 다르게 생성되어 가능한 적은수의 삼각요소가 사용되었다. Fig. 17(c)와 (d)에서는 트리밍 곡선이 존재하는 경우를 다루었으며 이경우도 곡률의 변화에 따른 적합한 크기의 삼각요소가 영역별로 생성되어 있음을 알 수 있다.

4.2 비교 분석

Fig. 18는 상용 CAD 소프트웨어로부터 얻은 결과와 본 연구에서 제시한 알고리즘을 적용하여 얻은 결과를 비교한 것이다. Fig. 18(a)에서 볼 수 있는 것처럼, 상용 CAD 소프트웨어에서 생성된 메쉬들은

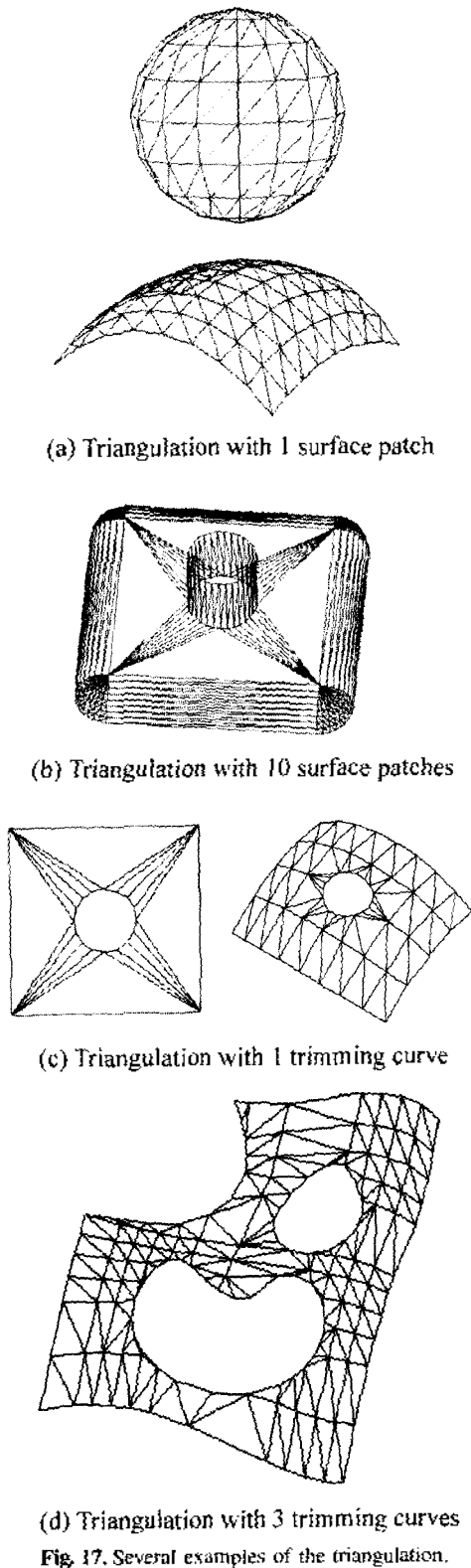
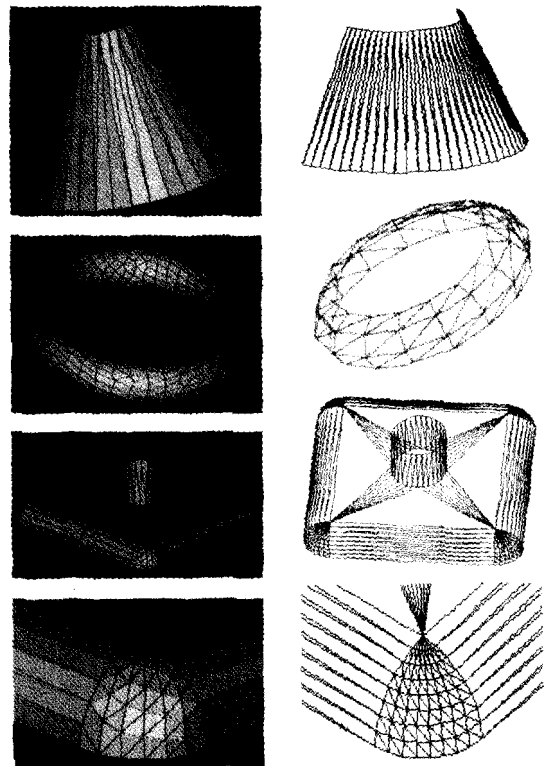


Fig. 17. Several examples of the triangulation.



(a) Result of a commercial CAD s/w

(b) Result of the proposed algorithm

Fig. 18. Comparison of the triangular mesh generation.

불규칙한 모양을 가지고 있다. 그와 반대로 제시된 알고리즘의 결과(Fig. 18(b))는 균일하게 생성된 메쉬들을 보여주고 있다.

아래의 결과들을 통해 알 수 있듯이 상용 CAD 소프트웨어의 결과(Fig. 18(a))보다 제시된 알고리즘(Fig. 18(b))으로 생성된 삼각 요소들이 균일한 것을 볼 수 있다.

따라서 공구경로나 STL파일을 생성했을 때 오차를 줄일 수 있다. 그 이유는 다음과 같다. 공구 경로나 STL파일은 삼각 다면체 모델과 평면의 교차점을 찾고 그것들을 연결하여 얻게 되는데, 삼각요소의 모양이 너무 예리(sharp)하면 교차점을 구했을 때 구멍이 생기거나 중복된 데이터가 나타날 수 있다. 또한, 불규칙적인 데이터를 이용해서 물체를 가공했을 경우에는 불규칙적인 표면이 생성된다. Fig. 18와 같은 예에서 상용 CAD 소프트웨어는 Fig. 19(a)와 같은 루프(loop)를 기반으로 공구 경로가 생성되기 때문에 실제 곡면과의 오차가 커진다. 하지만, 그와 반대로 본 연구에서 제시된 알고리즘을 적용한 결과는 Fig

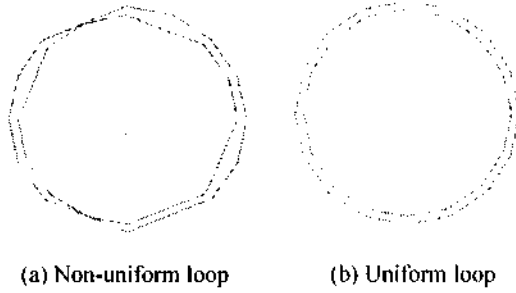


Fig. 19. Loop generation for a tool path.

19(b)와 같은 루프를 기반으로 공구경로가 생성되기 때문에 오차가 적게 발생된다.

5. 결론 및 추후 연구 과제

본 논문에서는 NURBS곡면이 매개변수로 표현되는 특징을 최대한 활용할 수 있도록 등매개 곡선을 이용하여 삼각메쉬를 생성하였다. 제시된 알고리즘을 몇 가지 사례에 적용하여 vertex to vertex 규칙을 따르는 유효한 삼각메쉬를 얻을 수 있었다. 삼각메쉬의 모양뿐만 아니라 곡면과의 공차도 중요한 의미를 갖게 되는데, 제시된 알고리즘을 이용하면 공차에 따라 삼각요소의 크기에 대한 조절도 가능하게 된다. 그 이외에도, 곡률을 고려하여 삼각 요소를 생성하기 때문에 평면 또는 곡률반경이 큰 부분에서는 필요 이상으로 작은 삼각 요소가 생성되지 않아서 요소의 개수가 감소될 수 있었다.

사각 곡면 패치에서는 모양이 균일하면서 유효한 삼각메쉬를 생성하여 공구 경로 생성이나 STL 파일 생성 등에서 발생할 수 있는 오차를 줄일 수 있었다. 하지만 트림 곡면에 대한 삼각화의 경우, 트리밍곡선의 모양이 너무 예리하거나(sharp) 여러 개의 트리밍 곡선들이 가까이 인접해 있으면 문제가 발생하는 경우가 존재한다. 이러한 문제점들은 추후 보완되어 더욱 다양한 모델에 대해서 알고리즘이 적용될 수 있도록 연구되어야 할 것이다.

감사의 글

본 논문은 동국대학교 교내연구비의 지원을 받아 수행된 연구결과이며 이에 감사드립니다.

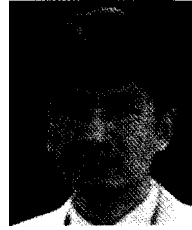
참고문헌

1. Cavendish, J. C., "Automatic triangulation of arbitrary planar domains", *International Journal for Numerical Methods in Engineering*, 8(4): pp. 679-696, 1978.
2. Hamann, B. and Tsai, P. Y., "A tessellation algorithm for the representation of trimmed NURBS surfaces with arbitrary trimming curves", *Computer Aided Design*, Vol. 28, No. 6/7, pp. 461-472, 1996.
3. Lee, Y. G., "Shape Model Data Conversion between Real Space Modeling Systems and Parametric Space Modeling Systems", '98한국 CAD/CAM학회 학술발표회 논문집, pp. 39-44, 1998.
4. Piegl, L. A., "On NURBS : A Survey", *IEEE Computer Graphics and Applications*, Vol. 11, No. 1, 1991.
5. Piegl, L. A. and Tiller, W., *The NURBS Book*, Springer, Berlin, 1995.
6. Piegl, L. A., and Richard, A. M., "Tessellating trimmed NURBS surfaces", *Computer Aided Design*, Vol. 27, No. 1, pp. 16-26, 1995.
7. Piegl, L. A. and Tiller, W., "Geometry-based triangulation of trimmed NURBS surfaces", *Computer Aided Design*, Vol. 30, No. 1, pp. 11-18, 1998.
8. Schumaker, L. L., "Triangulations in CAGD", *IEEE Computer Graphics and Applications*, Vol. 13, No. 1, pp. 47-52, 1993.
9. Sheng, X. and Hirsch, B. E., "Triangulation of trimmed surfaces in parametric space", *Computer Aided Design*, Vol. 24, No. 8, pp. 437-444, 1992.
10. Watson D. F., "Computing the n-dimensional tessellation with application to Voronoi polytopes", *The Computer Journal*, Heyden & Sons Ltd., Vol 2, No. 24, pp.167-172.
11. 구미정, 김동수, 전치수, 박세형, "곡면 트리밍", '97 한국 CAD/CAM학회 학술발표회 논문집, pp. 221-226, 1997.
12. 이진우, 컴퓨터 그래픽과 CAD, 영자문화사, 1995.
13. 정중환, 채수원, "트림 곡면에서 사각형 유한 요소망의 생성", '98 한국 CAD/ CAM학회 학술발표회 논문집, pp. 165-170, 1998.



정재호

1997년 한성대학교 산업공학과 학사
 1999년 동국대학교 산업공학과 석사
 관심분야: Surface Modeling, Geometric Algorithms, Rapid Prototyping



박준영

1982년 한양대학교 기계공학과 학사
 1985년 University of Minnesota 산업공학과 석사
 1991년 University of Michigan 산업공학과 박사
 1992년~1999년 한성대학교 산업공학과 조교수
 1995년~현재 동국대학교 산업공학과 부교수
 관심분야: Geometric Modeling, Computer Graphics, Mesh Generation, STEP