

경량 모델의 삼각 요소망으로부터 경계 곡선 추출 방법

권기연[†]

금오공과대학교 산업공학부

A Boundary Curve Extraction Method using Triangular Elements of a Lightweight Model

Ki-Youn Kwon[†]

School of Industrial Engineering, Kumoh National Institute of Technology

Received 11 October 2016; received in revised form 23 November 2016; accepted 1 December 2016

ABSTRACT

Sharing of CAD data plays a key role in the PLM and a lightweight model is widely used for visualizing and sharing a large data. The lightweight model is mainly composed of triangular elements to minimize file size. There is no problem at all to visually confirm the shape based on these triangular elements but there is a limit to numerically calculate the exact position on the curve or surface. In this paper, a boundary curve generation method using triangular elements is proposed to increase the utilization of lightweight models. After matching connectivity of triangular elements, boundary element edges are extracted. Boundary curves are generated by connecting of these boundary element edges. This proposed method has been tested on several models to demonstrate the feasibility.

Key Words: B-rep curve, Lightweight model, Product lifecycle management, Triangular element

1. 서 론

제품의 효율적인 제조를 위해 설계 단계에서부터 제품 폐기단계까지 데이터를 관리하는 PLM (product lifecycle management) 시스템이 활용되고 있다. PLM은 CAD 데이터 기반으로 구성되어 있으며, 각 단계에서 CAD 데이터에 대한 원활한 공유가 중요하다. 현재 기계, 조선 및 플랜트에서 사용되고 있는 CAD 시스템은 고가의 제품으로 많은 사용자가 이용하기 위해서는 비용이 발생한다. 또한 CAD 시스템에서 생성된 데이터는 형상의 기

하 정보뿐만 아니라 제품 제조에 필요한 많은 정보를 가지고 있어 데이터 용량이 커서 파일 공유 및 데이터 가시화에 효율성이 떨어지는 경우가 많다. 특히 수백만 이상의 부품으로 만들어지는 선박, 플랜트의 경우 CAD 데이터의 용량이 매우 크며, 최근 들어서 대용량 데이터에 대한 수요가 증가하고 있어 이와 같은 이슈는 커지고 있다. 대용량 데이터를 처리하기 위해서는 데이터의 크기를 줄이고, 가시화를 빠르게 할 수 있는 것이 필요하다. 이를 위해 경량 파일 포맷(lightweight file format)이 사용되고 있으며, 대표적인 파일 포맷으로는 JT, 3D XML, XVL, VRML 등이 있으며, 삼각망 정보만을 저장하고 있는 STL 파일도 경량 모델로 활용될 수 있다^[1,2]. Ball 등^[3] 및 Ding 등^[4]은

[†]Corresponding Author, mrkky@kumoh.ac.kr
©2017 Society for Computational Design and Engineering

경량 CAD 파일의 특징 및 주요 경량 CAD 모델 파일 포맷에 대해서 비교하였다. 경량 파일은 주로 형상 정보를 삼각망으로 저장하고 있으며, 경우에 따라서는 경계 곡선 등 일부 기하 정보를 가지고 있는 경우가 있다. 그러나 이와 같은 기하 정보가 파일 용량을 증가시키고, 처리 속도를 저해해서 최근에는 삼각망 정보만을 저장해서 주로 사용하고 있다. 기하 정보에 대한 중립 파일 포맷으로는 IGES, STEP이 많이 활용되고 있는데, Fig. 1의 형상을 IGES 파일로 저장한 경우 파일 크기가 228 Kbytes이며, 삼각망으로 이루어진 STL파일로 저장한 경우는 20 Kbytes이다. 두 파일의 저장 및 압축 방식에 따른 영향이 크지만, STL 파일은 삼각망을 더 densely하게 생성하면 훨씬 적은 파일 사이즈를 가질 수 있다. 또한 IGES 파일은 기하정보를 NURBS 곡선, 곡면으로 데이터를 저장하고 있어서 가시화를 위해 파일을 읽고, 삼각망으로 구성하는데 시간이 소요된다.

순수하게 삼각망으로 이루어진 경량 파일은 형상 가시화를 위해서는 최적이지만, 정확한 형상 정보가 필요한 경우에는 적용시킬 수 없다. Fig. 2는 선박 블록을 제작한 후, 설계 치수와 제작된 치수 간에 오차를 분석한 예이다. 이는 형상으로부터 추출된 설계 좌표와 3차원 계측기를 이용해서 측정

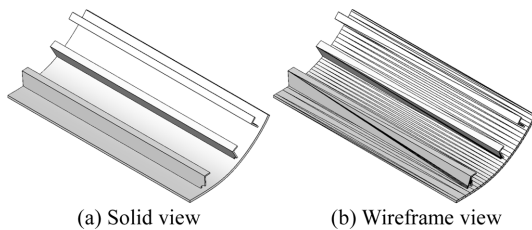


Fig. 1 A sample of lightweight file

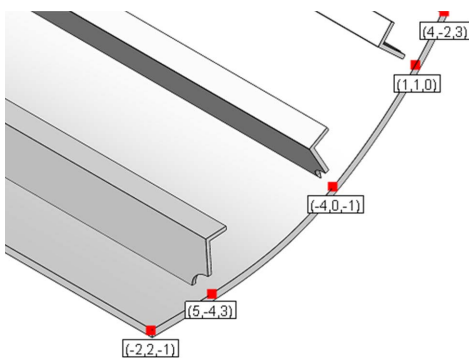


Fig. 2 Dimensional analysis of a large block of a ship

한 점을 비교 분석하여 오차를 분석한다^[5,6]. 이와 같은 경우 부재 끝 단의 정확한 설계 좌표가 필요하다. 부재에 곡률이 있는 경우 경량 모델의 삼각망을 이용해서 추출하면 오차가 존재한다. 삼각망으로 곡면을 추출해서 경계 곡선을 이용할 수 있는데, 자동으로 곡면을 생성하는 것은 많은 시간이 소요되며, 형상이 복잡한 경우에는 실패하는 경우가 많다. 따라서 본 연구에서는 이를 해결하기 위한 방법으로, 다양한 종류의 경량 모델로부터 정확한 경계 곡선을 안정적으로 추출할 수 있는 방법을 제안한다.

일반적으로 완성품으로부터 설계 형상을 이끌어내는 방식을 역공학(reverse engineering)이라고 한다. 측정 장치의 발전과 더불어 많은 발전이 이루어졌으며, 실제 제품 설계 및 생산 단계에서도 많이 활용되고 있다. 역공학은 측정점으로부터 다면체(polygon) 모델을 얻는 과정과, 다면체 모델로부터 곡면을 복원하는 과정이 있다. 대용량 측정점으로부터 다면체 모델을 얻는 과정은 상당 부분 자동화가 이루어졌으나^[7-9], 곡면 모델을 얻는 과정은 매끄러운 경계를 생성하기 위해 많은 수작업이 요구된다. 다면체 모델은 주로 삼각 요소망이 사용되고 있으며, 영역을 어떻게 분할할지에 따라 결과가 달라지며, 최종적으로 매끄러운 곡면 생성을 위해서는 많은 후처리 과정이 필요하다^[10-12]. 일반적으로 선박 건조에 활용되는 CAD 형상은 라운딩 처리가 되어 있지 않아 비교적 경계 구분이 명확하다. 본 연구에서는 많은 계산 시간이 소요되는 곡면 복원 단계를 거치지 않고, 선박 건조에서 활용되는 경량 파일로부터 빠른 시간 안에 경계 곡선을 추출할 수 있도록 하였다.

논문은 다음과 같이 구성되어 있다. 2절에서는 경량 모델 배경 지식과 다양한 파일 구조에 대해서 설명한다. 3절에서는 본 연구에서 제안하는 경량 모델의 삼각망으로 경계 곡선을 생성하는 방법에 대해 설명한다. 4절에서는 경계 곡선 추출을 적용한 실험 및 검증 결과를 살펴보고, 5절에서는 결론 및 향후 연구 방향에 대해서 논의한다.

2. 경량 파일 및 삼각망 구조

Fig. 3(a)는 경계 곡선으로 표현된 CAD 형상을 나타내고 있으며, Fig. 3(b)는 이를 삼각망으로 생성(tessellation)한 모습이다. 삼각망으로 이루어진

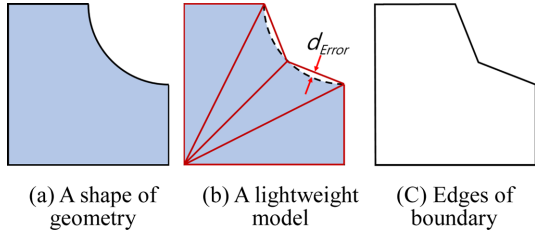


Fig. 3 Tessellation

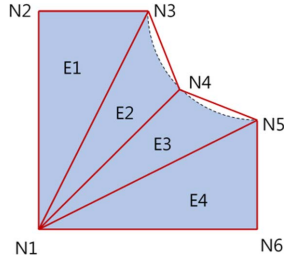


Fig. 4 Triangular elements of lightweight file

경량 모델을 가시화를 하면 최종적으로 Fig. 3(c)와 같이 사용자한테 보여지게 된다. 여기서 경계가 곡선으로 이루어진 경우 많은 수의 삼각형으로 분할될 때 실제 형상과 가까워 지게 된다. 그러나 너무 조밀하게 분할되는 경우는 데이터가 많아져서 메모리 사용량이 증가되며, 가시화 성능을 저하시킨다.

경량 모델의 삼각망을 저장하는 자료 구조는 일반적으로 다음 3가지 방법이 사용된다. 첫 번째 방식은 요소를 요소가 이루는 절점의 좌표로 표현하

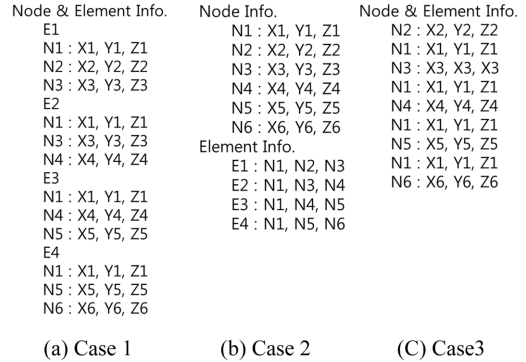


Fig. 5 Data structure of triangular elements

는 방식이다. Fig. 4는 요소 4개와 절점 6개가 존재하는 경우이다. 첫번째 방식은 이를 표현하기 위해 요소 4개에 해당하는 절점의 좌표 값을 각각 적어주는 것이다(Fig. 5(a)). 두 번째 방식은 절점 6개에 해당하는 좌표 값을 모두 적어주고, 요소가 가리키는 절점의 번호를 적어준다. 세 번째 방식은 절점을 순서대로 연결해서 삼각형을 스트립 구조로 표현하는 방식이다. 경량 모델 파일 중에서 STL 파일은 첫 번째 방식이며, VRML 파일은 두 번째 방식이다. 그리고 JT 파일은 세 번째 방식을 이용하고 있다. Fig. 4의 경우에는 (N2, N1, N3), (N1, N3, N1), (N3, N1, N4), (N1, N4, N1)...와 같은 삼각형으로 표현되며, (N1, N3, N1)과 같은 불필요한 요소가 표현되기도 한다. 3가지 방식에서 절점과 요소 정보 표현에 이용된 메모리 사용량은 Table 1과 같이 삼각형을 스트립으로 표현하는 방

Table 1 Memory usage of lightweight file formats

| | Case 1 | Case 2 | Case 3 |
|---------------------|--|--|--|
| 절점, 요소 정보 | - X, Y, Z 좌표 값 12개: float 타입변수 36개 4bytes*36 = 144bytes | - X, Y, Z 좌표 값 6개: float 타입변수 18개 - 요소 절점 번호 4개: integer 타입변수 12개 4bytes*18 + 4bytes*12 = 120bytes | - X, Y, Z 좌표 값 9개: float 타입변수 27개 4bytes*27 = 108bytes |
| 법선 정보 | -법선 벡터(X, Y, Z) 4개: float 타입변수 12개 4bytes*12 = 48bytes | -법선 벡터(X, Y, Z) 6개: float 타입변수 18개 4bytes*18 = 72bytes | - 법선 벡터(X, Y, Z) 9개: float 타입변수 27개 4bytes*27 = 108bytes |
| 전체 (절점, 요소 및 법선) | 192bytes | 192bytes | 216bytes |
| 경계 곡선 정보 | - 곡선 5개의 절점 연결 정보 (1, 2), (2, 3), (3, 4, 5), (5, 6), (6, 1): integer 타입변수 11개 - 절점 6개에 대한 X, Y, Z 좌표 값: float 타입변수 18개 4bytes*11 + 4bytes*18 = 116bytes | | |

식(case3)이 가장 적다. 또한 이와 같은 삼각망을 가시화 할 때 주로 OpenGL을 사용하는데 GL_TRIANGLE_STRIP을 사용하여 화면에 효율적으로 표현이 가능하다. 첫 번째와 두 번째 경우는 GL_TRIANGLES를 이용하여 가시화하여 첫 번째에 비해서는 가시화 성능이 떨어진다. 그러나 가시화를 정확하게 하기 위해서는 절점의 법선 정보가 필요한데, 세 번째 경우에는 중복된 절점을 포함해서 각 절점마다 법선 정보를 추가적으로 가지고 있어야 해서 전체적으로는 메모리 사용량이 가장 많을 수 있다. 두 번째 경우에는 중복된 절점이 없어서 절점의 인접한 요소를 이용해서 법선 방향을 계산할 수 있어서 법선 정보를 파일에 가지고 있을 필요는 없다. 저장 파일 크기 및 메모리 사용 측면에서는 두 번째 경우가 가장 효율적이고, 가시화 측면에서는 세 번째 경우가 가장 효율적이다. 그리고 삼각망으로 이루어진 모델에서 경계를 추출하기 위해서는 각 요소간의 연결 정보를 생성하는 것이 중요한데, 첫 번째와 세 번째 경우는 중복 절점이 존재해서 이를 제거하는 과정이 필요하다. 궁극적으로는 두 번째와 같은 자료 구조를 만드는 것이 필요하며, 다음 장에서 설명할 예정이다.

그리고 경량 파일에 경계 곡선 정보를 저장한다면 Fig. 3의 예제는 경계 곡선이 5개이고, 경계에 해당하는 포인트가 6개가 있어서 최소 Table 1과 같이 116 bytes가 증가하여 법선 정보를 제외한 파일 메모리 사용량과 유사할 만큼 많은 비중을 차지한다. 따라서 실시간으로 경계 정보를 복원할 수 있다면 훨씬 효율적으로 경량 파일을 활용할 수 있다.

3. 경량 모델 기반 경계 곡선 생성 방법

3.1 경계 곡선 생성 절차

경량 모델은 2절에서와 같이 삼각망으로 구성되어 있으며, 요소의 절점의 위치가 Fig. 4에서와 같이 실제 CAD형상 위에 있다. 절점과 절점 사이의 위치는 실제 CAD형상과 차이가 발생하므로, 본 연구에서는 다음과 같이 경계에 있는 절점을 찾아서 3차원 B-spline 곡선을 만드는 방법을 제안한다. Fig. 5의 자료 구조를 가지는 경량 파일은 절점의 위치 값이 동일하더라도, 각 요소마다 다른 절점을 생성하는 경우가 발생한다. 요소 에지(edge)의 경계와 내부를 구분하기 위해서는 요소의 연결 정보를 생성하는 것이 필요한데, 이를 위해서는 중

복 절점을 제거해서 동일 위치에 하나의 절점을 생성하고, 이를 참조하는 요소가 동일 절점을 가리키도록 해야 한다. 즉 Fig. 5의 두 번째 자료 구조를 가지도록 한다. 경계 곡선을 추출하기 위해 다음과 같은 방법을 제안한다.

- 중복 절점 제거: 동일한 좌표 값을 가지는 절점이 여러 개 존재한다면, 한 개만 남기고 이를 제거한다. 제거된 절점을 참조하는 요소의 절점 정보를 남은 한 개의 절점으로 업데이트 한다.
- 자료 구조 생성: 요소 에지 자료 구조를 추가로 생성하고, 절점과 요소 에지, 요소 에지와 요소를 연결시켜 준다. 또한 절점에는 해당 절점을 참조하는 요소 정보가 모두 포함되도록 한다.
- 경계 에지 추출: 요소 에지에 인접한 요소가 존재하지 않거나, 두 요소가 이루는 각도가 큰 경우 경계 에지로 분류한다.
- 경계 곡선 생성: 요소 에지를 순차적으로 연결해서 경계 곡선을 생성한다. 요소 에지에 절점과 요소 정보가 저장되어 있어서 인접한 요소 에지를 찾을 수 있으며, 인접한 요소 에지와 각도가 유사한 경우에 하나의 곡선을 생성하기 위해 연결한다. 그렇지 않은 경우는 새로운 곡선이 생성되도록 한다.
- 경계 곡선 결합: 같은 위치에 존재하는 경계 곡선을 결합한다. 솔리드 형상의 경우 경량 모델을 위한 삼각 요소망 생성시에 곡면이 만나는 위치의 곡선에 분할 개수가 다를 경우 각각 경계 곡선이 생성될 수 있다. 이와 같은 경우 이를 연결시켜 준다.
- B-spline 곡선 생성: 요소 에지의 절점과 절점 사이는 실제 CAD 형상과 오차가 존재하는데, 이를 줄이기 위해서 3차 B-spline 곡선을 생성하여 사용한다.

3.2 중복 절점 제거

앞 절에서 설명한 바와 같이 Fig. 5의 (a), (c)와 같은 자료구조에서는 중복 절점이 존재한다. 파일 구조 내에는 절점의 관계를 나타내는 정보가 없으며, 순수하게 좌표 값을 이용해서 같은 위치에 존재하는지 판단해야 한다. 모든 절점에 대해서 동일 좌표인지를 체크하게 되면 계산 시간이 많이 소요되어 영역을 분할하여 인접해 있는 절점에 대해서만 중복 체크를 하는 방법이 사용될 수 있다. 그러나 영역을 나누는 자체도 별도의 자료 구조

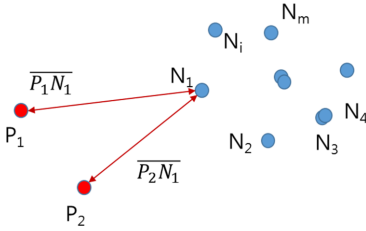


Fig. 6 Calculation of node distances

생성이 필요하며, 영역 안에 많은 절점이 존재할 경우 이에 대한 계산 시간 또한 다소 소요될 수 있다. 따라서 본 연구에서는 임의의 점에서 거리를 계산하고, 이를 거리 순으로 정렬하는 방식을 이용한다. 임의의 점을 하나만을 이용하는 경우는 위치가 달라도 거리가 비슷하게 나오는 경우가 발생할 수 있으므로 두 개의 임의의 점을 이용한다(Fig. 6). i 번째 절점에 대한 거리는 식 (1)과 같이 계산되며, 동일 거리를 가지는 경우 같은 위치에 있을 가능성이 있다. 이 경우에 대해서 두 절점의 거리를 다시 계산하여 같은 위치에 있으면 절점을 제거하고, 요소의 절점 정보를 업데이트 한다.

$$d_i = \overline{P_1N_i} + \overline{P_2N_i} \quad (1)$$

3.3 자료 구조 생성

본 연구에서는 형상의 경계 에지를 찾기 위해 Fig. 7가 같은 자료 구조를 사용한다. 요소(element)는 3개의 절점(node)과 에지(edge)를 가지고, 에지는 절점과 요소 정보를 가지고 있다. 각 절점에는 요소 정보를 가지고 있으며, 인접해 있는 모든 요소의 정보를 리스트로 저장한다. 에지 또한 요소 정보를 가지고 있으며, 인접해 있는 모든 요소의 정보를 리스트로 저장한다. 에지를 생성할 때는 에지를 구성하는 두 절점으로 이루어져 있는 에지가

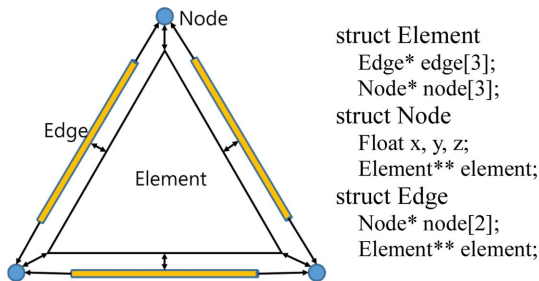


Fig. 7 Basic data structure for triangular element

존재하는지를 판단해서 없는 경우 새롭게 생성하고, 두 절점으로 이루어져 있는 에지가 존재하는 경우는 에지에 요소 정보만을 추가시켜 준다.

3.4 경계 에지 생성

3.3절과 같이 자료 구조가 생성되면 에지에는 주변 요소 정보가 저장되어 있다. 주변 요소 개수가 2개인 경우는 두 요소가 이루는 각도가 180도에 가까운 경우는 내부 에지로 분류하고, Fig. 8의 영역 ①이 같이 두 요소가 이루는 각도가 180도를 벗어나는 경우는 경계 에지로 분류한다. 본 연구에서는 임계 각도를 30도를 이용하였으며, 두 요소의 각도가 150도에서 210도 사이인 경우는 내부 에지로 분류한다. Fig. 8의 영역 ②와 같이 에지에 포함된 요소의 개수가 한 개인 경우는 경계 에지로 분류한다. 또한 에지에 포함된 요소 개수가 세 개 이상인 경우에도 경계 에지로 분류한다.

3.5 경계 에지 결합

3.4절을 수행하면 Fig. 9에서 표시된 영역에는 4개의 경계 에지가 생성된다. 에지에는 절점 정보가 저장되어 있어 인접한 경계 에지를 찾을 수 있다. 절점에서 인접한 두 경계 에지(boundary edge)가 이루는 각도가 180도에 가까운 경우에는 동일 경계 곡선상에 포함시킨다. 본 연구에서는 임계 각

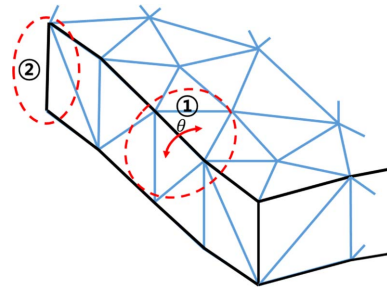


Fig. 8 Generation of boundary edges

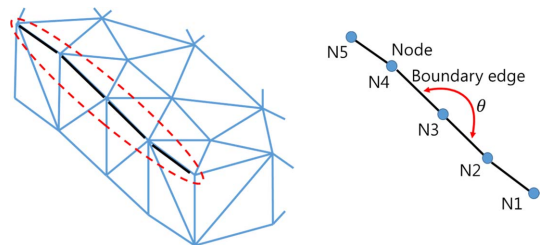


Fig. 9 Arrangement of boundary edges

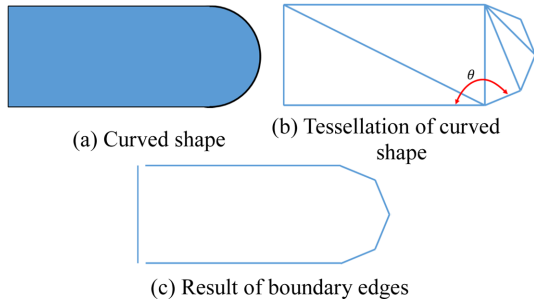


Fig. 10 Result of boundary edges on curved shape

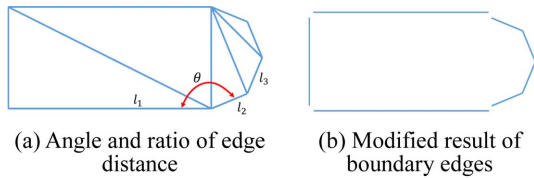


Fig. 11 Modified result of boundary edges on curved shape

도를 30도를 이용 했으며, 두 에지의 각도가 150도에서 210도 사이인 경우 동일 경계 곡선으로 분류한다. Fig. 9에서는 N1, N2, N3, N4, N5가 하나의 경계 곡선에 해당된다.

그러나 Fig. 10(a)와 같이 직선과 원호(또는 곡선)가 만나는 구간에서는 두 경계 에지가 이루는 각도가 180도에 가까워 Fig. 10(c)와 같이 하나의 경계 곡선으로 연결될 수 있다. 따라서 Fig. 11(a)와 같이 두 에지(l_1 , l_2)의 길이비의 차이가 크고, 원호(또는 곡선)의 시작 에지(l_2)와 다음 에지(l_3)의 길이비가 비슷한 경우에는 이를 연결하지 않는다(Fig. 11(b)).

3.6 경계 곡선 결합

형상에 가시화를 위한 삼각 요소를 생성할 때 인접한 곡면의 곡선에 대해 공유 정보가 없는 경우 Fig. 12와 같이 곡선에 생성된 요소의 개수가 달라 질 수 있다. Fig. 4에서 보듯이 요소의 절점 위치는 실제 형상과 정확히 일치하는 위치이다. 삼각 요소 정보로부터 더 정확한 경계 곡선을 복원하기 위해서는 요소의 절점들을 모두 이용하는 것이 필요하다. 곡선에 생성된 요소가 다른 경우는 시작과 끝점은 동일하지만 개수가 다른 경우(Fig. 13(a)), 경계 곡선이 다른 경계 곡선에 포함되는 경우(Fig. 13(b)), 그리고 경계 곡선들이 일부 중첩되는 경우(Fig. 13(c))가 있다. 첫 번째 경우는 두 경

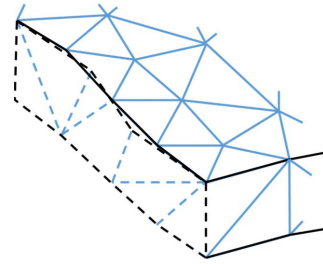


Fig. 12 Different node locations on boundary

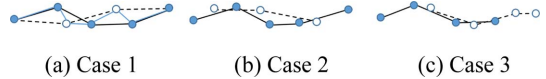


Fig. 13 Cases of different node locations

계 곡선의 시작점과 끝점이 같고, 곡선의 가운데에 놓인 점이 인접한 곡선 상에 있을 때이다. 두 번째는 길이가 작은 곡선의 시작점과 끝점이 인접한 곡선상에 있는 경우이며, 세 번째는 두 경계 곡선의 시작점과 끝점 중 한 개가 인접한 경계 곡선에 각각 포함되는 경우이다.

3.7 B-spline 곡선 생성

3.6절까지 생성된 경계 곡선은 3차원 포인트들의 집합으로 이루어져 있다. 포인트는 실제 형상 위의 점이지만, 포인트들 사이는 실제 형상과 오차가 존재한다. 이를 최소화 하기 위해 포인트들을 보간(interpolation)해서 3차 B-spline 곡선을 생성한다.

4. 실험 및 검증

제안하는 방법의 검증을 위해서 먼저 Fig. 14와 같은 형상에 대해 경계 곡선을 생성하는 실험을 수행하였다. 이는 Fig. 5의 첫 번째 파일 구조에 해당하며, 삼각형을 읽어서 중복 절점을 제거하고, B-spline 경계 곡선을 생성한 예이다. Fig. 15는 인접한 면들의 경계가 일치하지 않은 경우이다. 그림에서 네모로 표시된 부분이 위 면의 절점이며, 동그라미로 표시된 부분이 측면에 포함된 절점이다. 위 면에서 생성된 경계 곡선과 아래 면에서 생성된 경계 곡선을 결합해서 Fig. 15(b)와 같이 하나의 B-spline 곡선을 생성하였다. Fig. 16(a)는 구조물 지지대이며, 30,144개의 삼각형과 90,432개의 절점으로 구성된 모델이다. 실험 PC i7-6700에

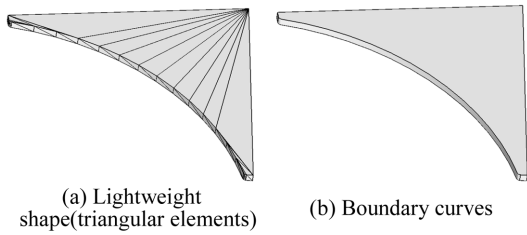


Fig. 14 Boundary curve generation for test case 1

서 중복절점 제거 한 후 최종 경계 곡선을 생성하기 까지 0.078초가 소요되었다. 최종적으로 절점의 개수는 중복 절점이 제거되어 29,706개이며, 790개의 경계 곡선이 생성되었다(Fig. 16(b)). Fig. 16(c) 는 상용 소프트웨어에서 곡면을 자동 생성한 것으로 4,953개의 곡면이 생성되었으며, 동일 PC에서 31초가 소요되었다. 또한 확대한 그림에서 보듯이 홀 주변이 매끄럽지 못하다.

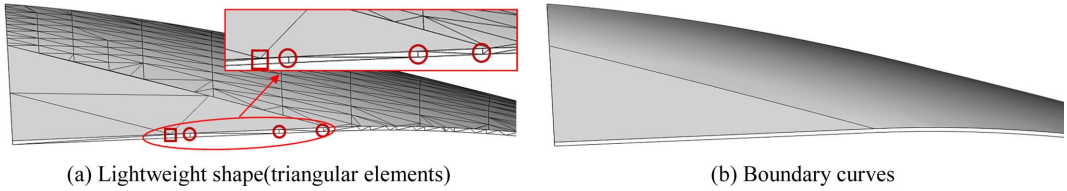


Fig. 15 Boundary curve generation for test case 2

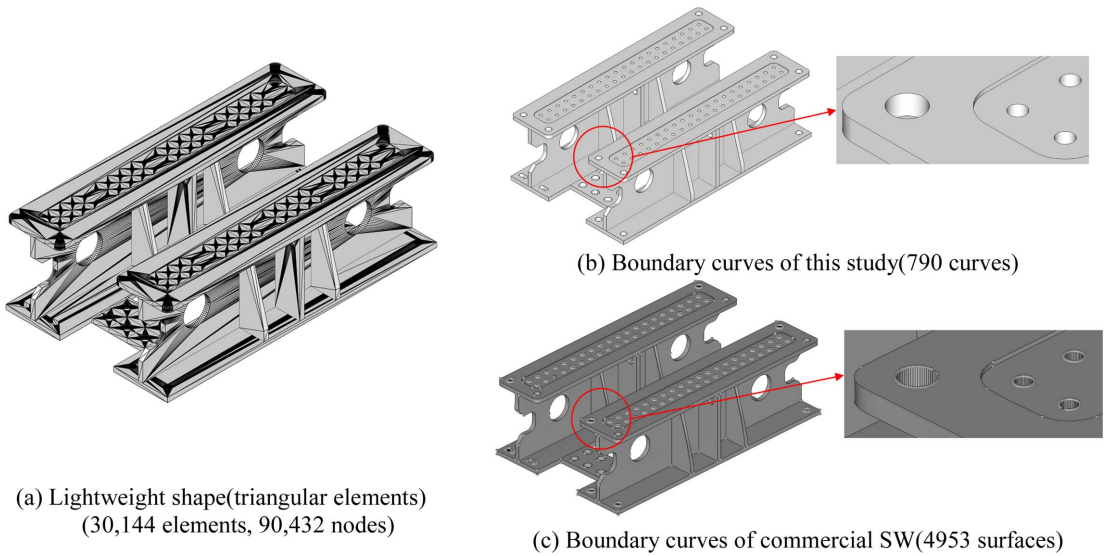


Fig. 16 Boundary curve generation for test case 3

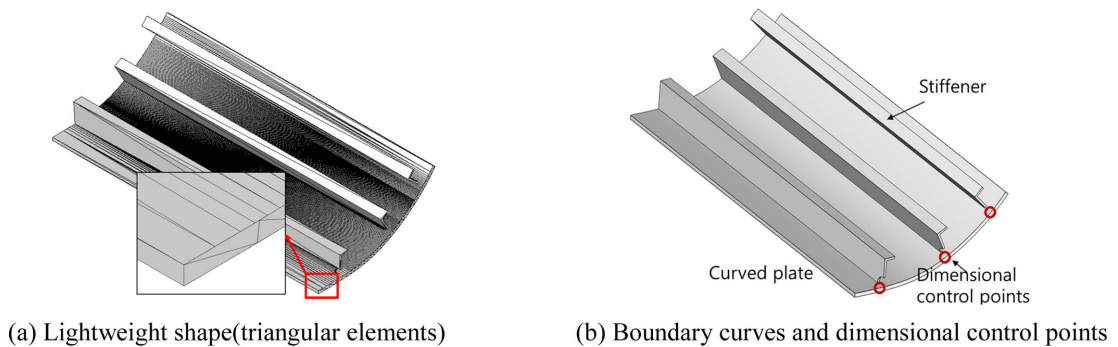


Fig. 17 Boundary curve generation for test case 4

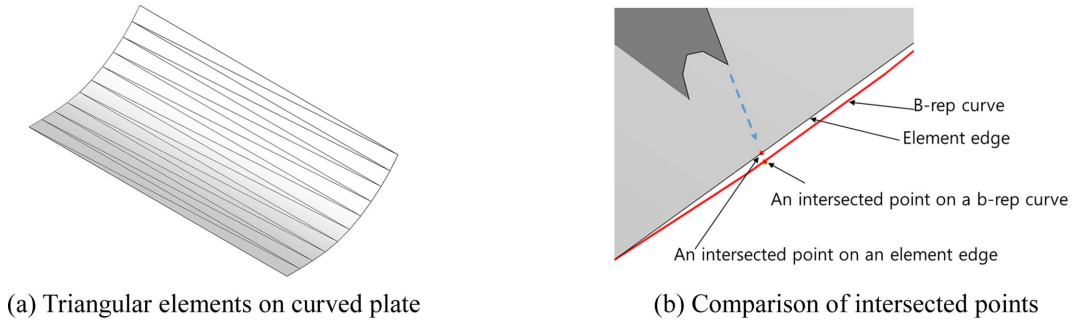


Fig. 18 Generation of dimensional control points

Fig. 17은 조선에서 곡블록을 제작 후 치수 오차 분석을 활용한 예이다. 경량 모델은 낮은 사양 PC에서 많은 양의 데이터를 가시화 할 수 있어 생산 지원 프로그램으로 많이 활용되고 있다. 곡블록은 곡주판(curved plate)과 보강재(stiffener)로 구성되어 있으며, Fig. 17(b)와 같이 곡주판과 보강재가 교차되는 위치가 치수 관리 포인트이다. 즉 3차원 계측기를 이용해서 해당 위치를 측정하고, CAD 형상으로부터 추출된 설계 점과 비교하여 치수 품질을 분석한다. 본 연구에서는 경량 모델을 이용하여 해당 설계 점을 추출하였다. 곡주판에 생성되는 요소 개수를 5, 10, 그리고 20개로 나누어서 실험을 하였다. Fig. 18(a)는 곡주판에 대해서 10개로 분할된 경우이다. 이를 Fig. 18(b)와 같이 요소 에지(element edge)와 교차된 점, 본 연구의 결과물인 경계 곡선(b-rep curve)과 교차된 점을 실제 CAD 형상에서 곡선을 이용해서 생성된 점과 비교하여 오차를 정리했다. Table 2에서와 같이 5개로 분할하였을 경우는 요소 에지와 CAD 형상의 곡선과의 차이를 나타내는 오차 값(chordal distance)이 6.40 mm에 해당하며, 본 연구의 알고리즘을 이용해서 경계 곡선을 추출한 결과와 실제 CAD 형상의 곡선과의 길이 차이가 0.0788 mm가 발생하였다. 그리고 CAD 형상을 이용하여 곡부재와 보강재와의 교차점과 본 연구에서 추출한 경계 곡선과 보강재와의 교차점의 최대 오차 값이

0.3983 mm가 발생하였다. 경계 곡선을 이용하지 않고, 삼각형의 요소 에지를 이용한 경우는 4.7907 mm가 발생하였다. 분할 개수 10을 준 경우에는 본 연구 결과와 CAD의 차이가 포인트 간의 최대 오차 값이 0.0335이고, 분할 개수를 20으로 준 경우에는 최대 오차 값이 0.0034가 나타났다. 치수 품질 관리를 위해 정확한 설계 값을 제공하는 것이 필요하며, 선박 건조에 있어서 관리 수준은 대조 블록(large block)인 경우 5~7 mm이다. 현재 선박 건조에 적용되고 있는 3차원 계측기의 정밀도 수준은 0.5 mm 정도이며, 정확한 관리를 위해서는 설계 값의 오차가 0.1 mm 이내로 만족하는 것이 필요하다. 곡부재의 형상에 따라 달라지지만, 대체적으로 경량 파일의 chordal distance 값을 3 mm 정도 수준으로 맞추면 설계 점을 0.1 mm 이내 오차를 가지도록 생성할 수 있다.

5. 결 론

본 연구에서는 다양한 종류의 경량 모델로부터 경계 곡선을 추출하는 방법을 제시하였다. 제안하는 방법은 삼각형의 연결성 정보를 생성하기 위해 중복 절점을 제거하고, 경계 에지 추출을 효율적으로 하기 위해 요소, 요소 에지, 절점으로 구성된 자료 구조를 생성하였다. 인접 요소 정보를 이용해서 경계에 존재하는 요소 에지를 추출하고, 이

Table 2 Comparison between CAD and b-rep curve, CAD and element edge

| Number of division (Chordal distance) | Curve length error (CAD & b-rep curve) | Point distance error (CAD & b-rep curve) | Point distance error (CAD & element edge) |
|--|---|---|--|
| 5 (6.40) | 0.0788 | 0.3983 | 4.7907 |
| 10 (1.45) | 0.0023 | 0.0335 | 1.3096 |
| 20 (0.38) | 0.0022 | 0.0034 | 0.3017 |

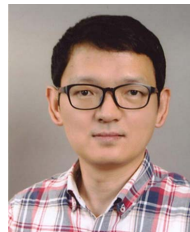
요소 에지를 순차적으로 연결하여 경계 곡선을 구성하였다. 또한 형상의 인접한 면에 대한 삼각형 요소 분할 개수가 다른 경우에는 동일 위치에 여러 개의 곡선이 생성될 수 있는데, 이를 연결하여 생성된 경계 곡선의 정확성을 높였다. 삼각망으로부터 곡면을 추출하는 방식은 계산에 많은 시간이 소요되며, 경계를 매끄럽게 추출하는 데는 경우에 따라서 많은 오류가 존재한다. 그러나 본 연구에서 제안하는 방법을 이용할 경우 빠른 시간 안에 정확한 경계 곡선을 추출할 수 있었다. 제안하는 방법을 검증하기 위해 선박의 부재를 이용해서 검증을 하였으며, 경량 모델을 기반으로 정확한 치수 품질 관리를 할 수 있도록 정구 점을 오차 없이 생성할 수 있도록 하였다. 이는 생산 지원 프로그램에서 많이 활용되는 경량 모델 파일 구조를 그대로 이용하여 CAD 모델의 경계 곡선을 오차 없이 복원함으로써 경량 모델에 대한 활용도를 높일 수 있다.

감사의 글

본 연구는 금오공과대학교 학술 연구비에 의하여 지원된 논문입니다.

References

1. Toriya, H., 2008, Benefits of Lightweight 3D Data. 3D Manufacturing Innovation, *Revolutionary Change in Japanese Manufacturing with Digital Data*, Springer, pp.33-50.
2. Cheon, S.U., Lee, J.H., Park, K.P. and Suh, H.W., 2013, Requirement Analysis on Lightweight CAD Models in Ship PLM Environment and Its Application Examples, *Transactions of the Society of CAD/CAM Engineers*, 18(4), pp.299-307.
3. Ball, A., Ding, L. and Patel, M., 2007, Lightweight Formats for Product Model Data Exchange and Preservation, *PV 2007 Conference*, pp.9-11.
4. Ding, L. and Ball, A., 2007, Product Representation in Lightweight Formats in Product Life-cycle Management, *4th International Conference on Digital Enterprise Technology*, UK.
5. Noh, J.K., 2013, An Algorithm for Optimized Accuracy Calculation of Hull Block Assembly, *Journal of the Korean Society of Marine Environment & Safety*, 19(5), pp.552-560.
6. Kwon, K.Y., 2016, A Weighted Points Registration Method to Analyze Dimensional Errors Occurring during Shipbuilding Process, *Transactions of the Society of CAD/CAM Engineers*, 21(2), pp.151-158.
7. Gálvez, A. and Iglesias, A., 2012, Particle Swarm Optimization for Non-uniform Rational B-spline Surface Reconstruction from Clouds of 3D Data Points, *Information Sciences*, 192, pp.174-192.
8. Lafarge, F. and Alliez, P., 2013, Surface Reconstruction through Point Set Structuring. *In Computer Graphics Forum*, Blackwell Publishing Ltd, 32(2.2), pp.225-234.
9. Berger, M., Tagliasacchi, A., Seversky, L.M., Alliez, P., Guennebaud, G., Levine, J.A. and Silva, C.T., 2016, A Survey of Surface Reconstruction from Point Clouds, *In Computer Graphics Forum*, DOI: 10.1111/cgf.12802.
10. Xiao, D., Lin, H., Xian, C. and Gao, S., 2011, CAD Mesh Model Segmentation by Clustering, *Computers & Graphics*, 35(3), pp.685-691.
11. Yan, D.M., Wang, W., Liu, Y. and Yang, Z., 2012, Variational Mesh Segmentation via Quadratic Surface Fitting. *Computer-Aided Design*, 44(11), pp.1072-1082.
12. Theologou, P., Pratikakis, I. and Theoharis, T., 2015. A Comprehensive Overview of Methodologies and Performance Evaluation Frameworks in 3D Mesh Segmentation, *Computer Vision and Image Understanding*, 135, pp.49-82.



권 기 연

1999년 고려대학교 기계공학과 학사
 2001년 고려대학교 기계공학과 석사
 2006년 한국과학기술원 기계공학과 박사
 2006년~2016년 삼성중공업 중앙연구소 차장
 2016년~현재 금오공과대학교 산업공학부 교수
 관심분야: CAD/CAE, Digital manufacturing, Dimensional quality management, Mesh generation, PLM, 3D measurement system