



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

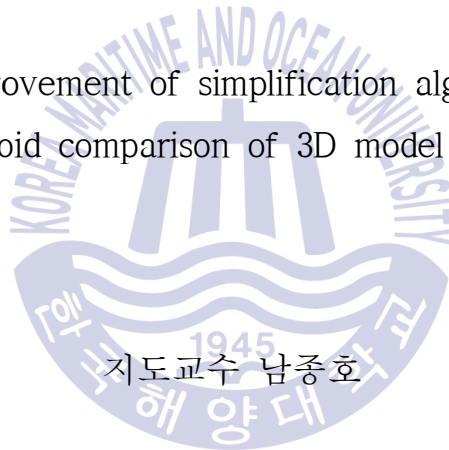
이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공학석사 학위논문

해양플랜트 전산 모델의
모서리 영역과 도심 비교를 활용한
경량화 알고리즘 분석과 개선

Analysis and improvement of simplification algorithm using edge
region and centroid comparison of 3D model of offshore plant



지도교수 남종호

2018년 2월

한국해양대학교 대학원

조선해양시스템공학과
이민규

本 論文을 李珉奎의 工學碩士 學位論文으로 認准함.

위원장 이 승 재 (인)

위 원 남 종 호 (인)

위 원 박 주 용 (인)



2017년 12월 5일

한국해양대학교 대학원

목 차

목차	ii
List of Tables	iv
List of Figures	v
Abstract	vii

1. 서 론

1.1 연구배경 및 필요성	1
1.2 국내외 기술 연구 현황	2
1.2.1 국외 연구 현황	2
1.2.2 국내 연구 현황	3
1.3 연구 목적	4

2. 연구 배경 이론

2.1 3D 모델 데이터 형식	5
2.1.1 NURBS	6
2.1.2 폴리곤 메쉬	7
2.2 Garland 알고리즘	8
2.2.1 Quadric Error Metrics	8
2.2.2 Garland 알고리즘 적용	10
2.3 Metro	11

3. Garland 알고리즘을 적용한 경량화 모델의 문제점 분석	15
3.1 Garland 알고리즘으로 나타난 경량화 모델의 외형적 오류	19
3.2 정점의 위치에 따른 모서리의 케이스 구분	21
3.2.1 깨지기 쉬운 모서리	23
3.3 경계 모서리의 역할과 구현	24
3.4 가중치가 모서리에 끼치는 영향의 범위	26
4. 도심 보존을 통한 경량화 개선 알고리즘	
4.1 깨지기 쉬운 모서리의 허용 영역 설정	
4.1.1 새로운 폴리곤 꼭짓점의 허용 영역	29
4.1.2 허용 영역 : 구	30
4.1.3 정당성 검토	31
4.2 깨지기 쉬운 모서리의 새로운 폴리곤 꼭짓점 선정	34
4.2.1 모서리의 경로에 존재하는 점을 이용한 새로운 폴리곤 꼭짓점 선택	34
4.2.2 폴리곤 그룹 설정과 비교	35
4.2.2.1 면적 비교	37
4.2.2.2 도심 비교	38
4.2.3 도심이 가진 기하학적 특성	39
5. 결론	
5.1 결과	42
5.2 향후 과제	43
References	45

List of Tables

Table 1	NURBS & Polygon mesh	8
Table 2	Comparison of distance using metro	13
Table 3	Comparison of ratios of outside edge	33
Table 4	Comparison of distance of both method using metro	41



List of Figures

Fig. 1 Complex shape of offshore plant	2
Fig. 2 3D model offshore plant	2
Fig. 3 A sequence of polygon mesh simplification	3
Fig. 4 NURBS Equation	7
Fig. 5 Distance from vertex to polygon by vector	10
Fig. 6 Process of Garland algorithm	11
Fig. 7 Distance between both polygon	12
Fig. 8 Error of both polygon	12
Fig. 9 Evaluate the sign of distance	13
Fig. 10 Model 1 for metro	14
Fig. 11 Model 2 for metro	14
Fig. 12 Original bunny model with 69,451 triangles	15
Fig. 13 Approximated bunny model with only 1,000 triangle meshes	15
Fig. 14 Original dehydrator model with 454,202 triangles	16
Fig. 15 Approximation dehydrator model with only 200,000 triangle meshes	16
Fig. 16 Original column model with 454,202 triangle meshes	17
Fig. 17 Approximated column model with only 200,000 triangle meshes ..	17
Fig. 18 Original low pressure tank model with 454,202 triangles	18
Fig. 19 Approximated low pressure tank model with only 200,000 triangle meshes	18
Fig. 20 Original radiator model with 454,202 triangles meshes	19
Fig. 21 Approximated radiator model with only 200,000 triangle meshes ·	19
Fig. 22 Simplification process	20
Fig. 23 A part isolated from original model	20
Fig. 24 Case with fatal error, isolated from original model	21

Fig. 25	Case of new vertex for polygon location	21
Fig. 26	Polygons in the yz plane	22
Fig. 27	Z values out of the min-max box	24
Fig. 28	A 3D model	25
Fig. 29	Q matrix calculation for edge	25
Fig. 30	Number of polygon connected on edge	26
Fig. 31	Weight affect on other edge connected to the boundary edge ...	27
Fig. 32	Examples of geometric spaces	30
Fig. 33	Allowable space set by a sphere	31
Fig. 34	Location analysis of new vertices	32
Fig. 35	10 points selected on edge	35
Fig. 36	Original polygon group and breakable edge	36
Fig. 37	Original polygon group and outermost edge	36
Fig. 38	Original model and one of case points on edge	37
Fig. 39	Original polygon group in 3D	38
Fig. 40	Cases of simplified model connect to a point in the path of an breakable edge	38
Fig. 41	Translation of centroid	39
Fig. 42	Result of area	39
Fig. 43	Result of centroid	39
Fig. 44	Error between original & simplified model after simplified using area variation	40
Fig. 45	Error between original & simplified model after simplified using centroid variation	41
Fig. 46	Algorithm for selection new vertex on edge	42
Fig. 47	Further simplification at higher levels	43

해양플랜트 전산 모델의 모서리 영역과 도심 비교를
활용한 경량화 알고리즘 분석과 개선

Analysis and improvement of simplification algorithm
using edge region and centroid comparison of 3D model
of offshore plant

Lee, Min Kyu

Department of Naval Architecture & Ocean Systems Engineering
Graduate School of Korea Maritime and Ocean University



Abstract

해양플랜트와 같은 대형 시스템은 그 크기가 매우 크고 복잡한 형상을 가지고 있어 대량의 3차원 데이터를 가진다. 이 거대한 3D 모델을 시뮬레이션하기 위해서는 경량화 알고리즘이 필수적이다.

해양플랜트 모델을 경량화하기 위해, 폴리곤 메쉬 모델의 경량화에 흔히 사용 되는 Garland 알고리즘을 해양플랜트에 적용하였다. Quadric error metrics를 사용하여 폴리곤들이 가지는 특성을 이용한 경량화 방법이다. 그러나 해양플랜트 모델에 이 알고리즘을 적용한 결과, 원본 형상과는 매우 큰 오차를 가지는 경량화 결과 모델을 산출하였다. 이 현상에 대한 원인으로, 경계를 유지하기 위해 추가된 가중치의 부작용과 영향 범위에 대해서 설명한다.

본 연구에서는, 가중치로 인해 크게 어긋난 새로운 폴리곤 꼭짓점을 가지는 깨지기 쉬운 모서리에 대해서 설명하고, 이에 대한 기준과 근거를 제시한다. 그 후에 깨지기 쉬운 모서리가 이루는 폴리곤 그룹의 기하학적 변화량을 근거로 하여, 새로운 폴리곤 꼭짓점의 다른 계산 방법을 제시한다. 폴리곤 그룹의 각 평면들이 경량화 전후에 가지는 도심의 변화량을 비교하여 새로운 폴리곤 꼭짓점의 위치를 선정하였다.

KEY WORDS: Offshore plant 해양플랜트; Massive 3D model 대용량 3D 모델 ; Mesh simplification 메쉬 경량화 ; Breakable edge 깨지기 쉬운 모서리 ; Allowable area 허용 영역.



해양플랜트 전산 모델의 모서리 영역과 도심 비교를
활용한 경량화 알고리즘 분석과 개선

Analysis and improvement of simplification algorithm
using edge region and centroid comparison of 3D model
of offshore plant

Lee, Min Kyu

Department of Naval Architecture & Ocean Systems Engineering
Graduate School of Korea Maritime and Ocean University



Abstract

Large systems such as offshore plants are very large in size and have a complex shape. Thus, that have a large amount of three-dimensional data. A simplification algorithm is necessary to simulate this huge 3D model.

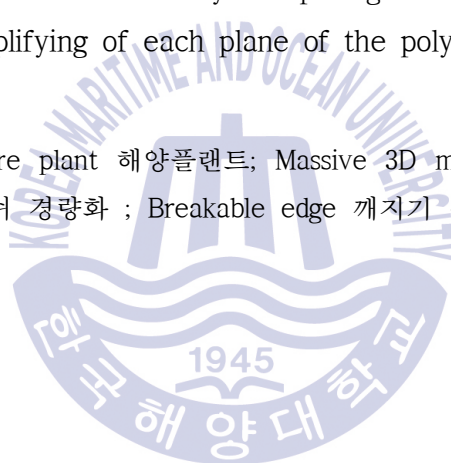
In order to simplified the offshore plant model, we applied the Garland algorithm, which is commonly used for lighter weight of the polygon mesh model, to the offshore plant.

It is a simplifying method using characteristics of polygons using quadric error metrics. However, applying this algorithm to the offshore plant model yielded a result with a very large error.

As a cause of this phenomenon, we describe the side effects and the range of influence added to keep the boundaries. Describe the side effects and range of edge weights added to maintain the boundary, which is the cause of this phenomenon.

In this paper, we describe a breakable edge with a new vertex that is largely offset by weight, and provide criteria and rationale for it. Then we present another calculation method of new vertex based on the geometric change of the polygon group formed by the breakable edge. The positions of the new vertex was selected by comparing the positions of centroid before and after simplifying of each plane of the polygon group.

KEY WORDS: Offshore plant 해양플랜트; Massive 3D model 대용량 3D 모델 ; Mesh simplification 메쉬 경량화 ; Breakable edge 깨지기 쉬운 모서리 ; Allowable area 허용 영역.



제 1 장 서 론

1.1 연구배경 및 필요성

국내 조선 산업은 1970년대에 세계 시장에 진입하여 2000년대 중반까지 급속히 성장하였다. 조선 대기업은 물론이고 중소 조선소와 조선 기자재 업체 등 관련 업체들도 호황을 유지하였다. 하지만 세계 금융위기 이후 세계 물동량이 급격하게 줄어들고, 중국과 인도, 필리핀 등에서 가격경쟁력을 내세우며 조선시장에 진출함에 따라 국내 조선업계에 큰 타격을 입히게 되었다. 국내에서는 이러한 선박의 경쟁력 약화에 대한 대비책으로 고부가가치선 시장 진출에 대한 필요성을 인식하고 해양플랜트 시장에 대한 비중을 지속적으로 늘려왔다.

해양플랜트의 건조영역에서 높은 점유율을 보이고 있으나, 그 외에 해양플랜트 서비스나 설계엔지니어링 등 다른 부분의 경쟁력은 매우 저조하여 해외에 의존하고 있다. 국내 조선업계가 세계 경쟁에서 우위를 유지하고 시장경쟁력을 강화하기 위해서는, 첫째로 지속적인 최신 기술 개발을 통해 기술적 우위를 점유하는 것, 둘째로 교육을 통해 질 높은 해양 전문 인력을 양성하는 것이 필요하다.

국내 대형 조선사를 중심으로 생산 프로세스에서의 기술 개발에 상당한 공을 들이고

있다. 특히 작업 현장에서의 휴대성이 좋은 모바일과 태블릿 플랫폼 환경으로의 확장을 위해 지속적으로 연구가 이루어지고 있다. 두 가지 환경 모두 하드웨어적 한계가 분명한 플랫폼이기 때문에, 용량이 매우 큰 해양플랜트 3D 모델을 가시화하기에 다소 부족한 능력을 가진다. 이를 해결하기 위한 기술 개발이 절실한 상황이다.

해양플랜트 인력 양성을 위한 교육 시스템도 이와 같은 맥락의 문제를 가지고 있다. 해양플랜트는 산업의 특성상 실제로 접할 기회가 매우 적기 때문에, 가상환경(Virtual environment)에 구축한 시뮬레이션 시스템을 이용하여 교육에 활용하는 경우가 많다.

그러나 해양플랜트의 경우 Fig. 1과 같이 운용 범위가 매우 넓고 복잡한 형상을 가지고 있다. 기반이 되는 설계 데이터 역시 Fig. 2로 볼 수 있는 것과 같이 대용량의 3차원 형상 데이터를 산출하게 된다. 이러한 데이터를 시뮬레이션에서 활용하게 되면 초기의 모델 로딩 시간도 매우 길다. 그 후에 행동입력이나 카메라의 위치 이동 등과 같은 명령을 수행하는 것은 시뮬레이션에 매우 큰 부담을 준다.

그렇기 때문에, 데이터를 시뮬레이션 시스템에 그대로 활용하는 거나 모바일 환경으로의 출력은 거의 불가능하기 때문에 일반적으로 간략화 과정을 수행한다. 이 과정에서 불필요한 폴리곤 메쉬를 제거하거나 리모델링하는 과정을 통해 대용량 해양플랜트 모델을 경량화 하고 있다.



Fig. 1 Complex shape of offshore plant

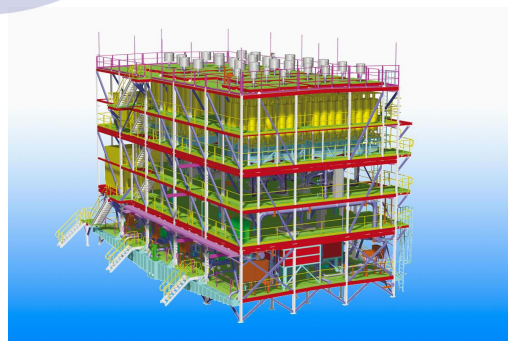


Fig. 2 3D model offshore plant

1.2 국내외 기술 연구 현황

3차원 모델 경량화 기술은 그래픽 품질과 속도가 중요시되는 3D 게임분야에서 활발하게 활용된다. 경량화를 위한 다양한 소프트웨어들이 판매되고 있고, open source library 들이 많이 개발되어 있다. 그러나, 경량화 수준을 결정하는 것은 활용되는 분야의 특성과 실무자의 경험에 의존하는 편이다.

1.2.1 국외 연구 현황

메쉬 경량화 방법은 거리오차(distance error)에 따라 정점을 연속적으로 제거하는 방법을 제시한 이후로 (William et al., 1992), 지금까지 계속 발전해 왔다. 폴리곤 메쉬 모델의 경량화는 모델 형상의 왜곡과 밀접한 관련이 있다.

Fig. 3 에서 볼 수 있는 것처럼, 계속 폴리곤 수를 줄인다면 형체를 알아보기 어려울 만큼 단순화 될 수도 있다. 이를 방지하기 위해, 모델 형상 왜곡 수준인 오차를 최소화 하면서 필요 이상의 폴리곤 개수를 줄이는 방법들이 연구되어 왔다. 대표적으로 에너지 최소화 방법을 이용하였으며(Hoppe, 1996), 이 후에 QEM이라는 오차 기준을 제시하였다 (Garland & Heckbert, 1997).

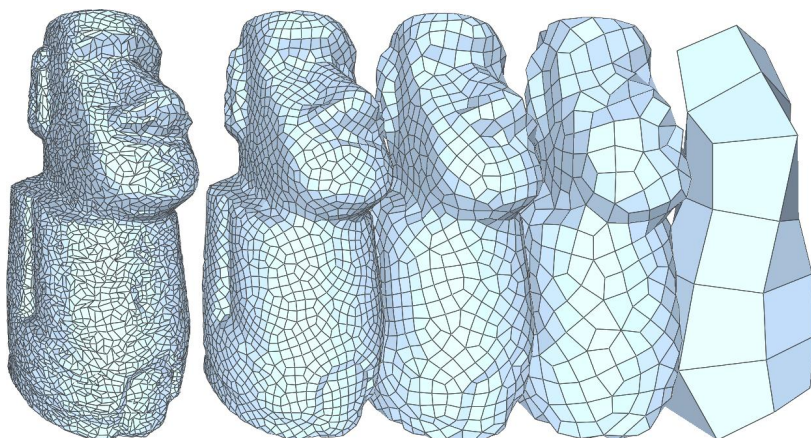


Fig. 3 Sequence of polygon mesh simplification

1.2.2 국내 연구 현황

기존 연구의 QEM기준을 기반으로 모델 형상의 오차를 효율적으로 줄일 수 있는 방안들이 제시되었다(Garland & Heckbert, 1997). 기술보고 내용을 보면(천상욱 등, 2009), 현재 활용중인 ‘Design Review System’에서는, CAD모델과 가시화 정보를 기반으로 하는 경량화 모델을 이원화하여 활용하고 있으며 대용량 모델의 가시화는 어려운 일임을 기술하고 있다.

시스템의 일부나 장비, 모듈 등에 대해 3D 모델 단순화 기법을 적용하려는 시도가 있으나(권순조 등, 2015), 아직 선박이나 해양플랜트와 같은 거대 시스템에 단순화 기법을 적용한 사례는 발표되지 않았다.

1.3 연구 목적

경량화 방법에 대한 다양한 연구가 이루어 졌지만, 해양플랜트의 특성을 고려한 경량화 방법이 매우 절실한 상황이다. 해양플랜트의 시뮬레이션의 성능 향상을 위해선 두 가지 접근 방법이 있다. 하나는 연산 능력을 향상시켜 원본 데이터를 그대로 사용 하는 방법이고, 다른 하나는 연산 능력보다 많은 데이터를 감축시키는 방법이 있다. 본 연구에서는 데이터를 감축시키는 방법에 대해서 접근하였다.

해양플랜트 시뮬레이션에 활용하기 위한 3차원 모델은 크게 두 가지로 구분할 수 있다. 3차원 모델 데이터를 수학적식에 기반 하여 공간에 표현하는 NURBS모델과, 점과 모서리를 이용하여 삼각형 혹은 사각형의 면의 조합으로 표현하는 폴리곤 메쉬 방법으로 나뉜다. 폴리곤 메쉬 방법은 좌표를 가지는 정점을 조합하여 수많은 면을 통해 3차원 모델을 표현하기 때문에, 수학적식을 이용한 NURBS보다 데이터의 크기가 작아서 연산 과정이 가볍다. 이러한 장점으로 데이터가 매우 큰 해양플랜트 3D 모델을 시뮬레이션에 활용할때 폴리곤 메쉬 방법이 많이 사용되고 있다.

폴리곤 메쉬의 경량화 방법으로 가장 많이 활용되고 있는 알고리즘을 활용하였다(Garland & Heckbert, 1997). 정점이 주변 평면과 가지는 오차를 이용한 모서리 삭제 방법

으로, 평면의 수를 정할 수 있는 경량화 방법이다. 이를 통해 해양플랜트 모델 형상의 경량화 작업을 수행하였다. 경량화 작업 후에 해양플랜트 모델에서 발생하는 극단적인 외형적 오류가 발생하는 것을 확인하였다. 경량화의 허용 영역을 통해 오류 케이스들을 구별하고, 폴리곤의 도심을 비교하여 경량화 폴리곤을 결정짓는 방법을 제시한다.



제 2 장 연구 배경 이론

2.1 3D 모델 데이터 형식

시뮬레이션에서 활용하는 3D 모델은 크게 2가지를 사용한다. 형상에 대한 곡선 및 곡면 정보를 수학적식으로 표현한 NURBS방법(Fig. 4)과 삼각형 또는 사각형과 같은 다각형 형태로 나타내는 폴리곤 메쉬 방법이다.

2.1.1 NURBS

NURBS는 Non-Uniform Rational B-Spline의 약자로 수학적인 식(Fig. 4)을 이용하여 곡선이나 곡면을 나타내는 방법(Piegl & Tiller W, 1997)이다. 3차원 기하모형을 수학적으로 나타낼 수 있는 방식 중 하나이며, 2·3차원에서의 곡 성분을 매우 정확하게 표현할 수 있다(Fig. 4). 유연성과 정밀성이 매우 뛰어나고 수학적으로 매우 안정적이기 때문에, 상용 CAD 프로그램(Rhino, NX, 3D Max, etc.)과 국제 표준 포맷(IGES, STEP)에서 사용되고 있다.

$$C(u) = \sum_{i=1}^k R_{i,n}(u)P_i$$

k : The number of control point
 P_i : Control point
 $R_{i,n}$: Basis function
 u : Parameter
 n : Degree

Fig. 4 NURBS equation

NURBS는 정밀하고 안정적이라는 장점을 가지고 있지만, 수학식에 기반을 두는 방법이라는 점에서 모델의 특성에 따라 계산과정이 기하급수적으로 증가하는 현상이 발생한다. 특히 지속적으로 시점의 변화가 이루어지는 시뮬레이션의 경우, 프레임단위로 NURBS를 계산해 줘야 한다는 큰 단점이 발생한다. 시뮬레이션에서 계산 량이 증가하게 되면 원하는 FPS(Frames per second)를 충족시키기 어렵고, 실시간 반응 또한 기대하기 어렵게 되면서 시뮬레이션의 현실감을 많이 떨어트리게 된다. 해양플랜트 3D 모델의 경우 매우 복잡한 구조를 가지고 있으므로, 이 모든 요소들을 NURBS로 표현하는 것은 시뮬레이션으로 활용하는 데엔 큰 한계를 가진다.

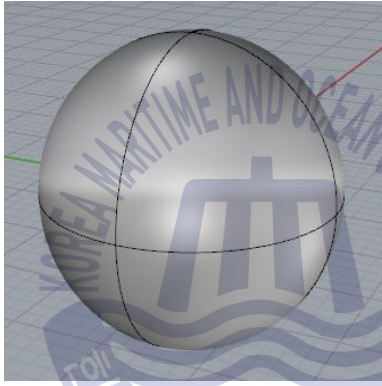
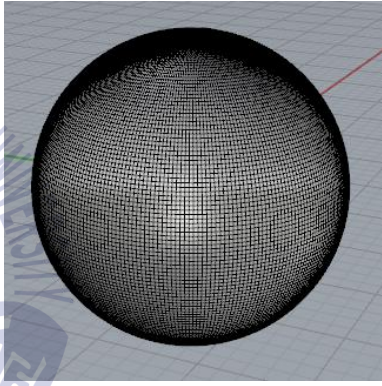
2.1.2 폴리곤 메쉬

폴리곤 메쉬 모델은 3D 그래픽스에서 정점(vertex), 모서리(edge), 평면(face)들의 집합으로 모델의 형태를 다각형으로 표현하는 방법이다. 가장 작은 단위인 정점들을 이어서 평면을 생성한다. 평면들의 집합으로 모델을 표현하기 때문에 NURBS와 달리 각 평면마다 질감을 표현할 수 있는 텍스처 매핑이 가능하며, NURBS보다 현실감 있는 렌더링을 할 수 있다.

또한 좌표 기반의 정점과 연결 순서를 이용한 모서리, 평면 데이터를 표현하므로 NURBS에 비해 데이터의 크기가 가볍다. 이러한 장점들로 많은 시뮬레이션에서 폴리곤 메쉬 형태의 3D 모델을 사용하고 있다. 같은 모델을 표현하더라도 폴리곤의 수가 늘어나

면 높은 하드웨어 성능을 요구하므로, 전체적인 형상을 유지하면서 모델의 특성을 고려하여 최적화 하는 것이 필요하다.

Table 1 Comparison of NURBS and polygon mesh

	NURBS	Polygon mesh
Usage	Modeling(CAD)	Rendering, Simulation
Element	Subdivided surface elements	(triangular) Polygons
Advantage	Exquisite modeling	Realistic Rendering (Texture mapping)
Example		

2.2 Garland 알고리즘

이번 연구에서 사용되는 경량화의 방법에 기초가 되는 Garland 알고리즘은 정점별로 가진 인근 평면까지의 거리 값을 이용한 경량화 파라미터(이하 오차)를 생성하여 메쉬의 수를 줄여 나가는 방식이다. 오차를 이용하여 모서리를 삭제하고 새로운 정점을 계산한 후 그 정점에 모서리를 재연결하는 방식을 반복적으로 실행하여 순차적으로 모서리를 삭제해 나가는 개념이다. Garland 알고리즘의 가장 큰 강점은 속도가 빠르고, 사용자가 경량화 하고자 하는 목표 폴리곤 수를 정할 수 있다는 점이다.

2.2.1 Quadric Error Metrics

앞서 설명하였던 경량화 파라미터를 계산하기 위해서, Garland 알고리즘에서는 ‘각 정점에서 해당 정점을 포함하는 평면들까지의 거리제곱 값들의 합’을 오차라고 정의하고 삭제 순서를 결정한다.

한 점(식(1))에서 평면(식 (2))까지의 거리 공식(식 (3))은 아래와 같다.

$$\vec{v} = \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} \quad (1)$$

$$\pi = ax + by + cz + d = 0 \quad \left(\vec{n} = \begin{pmatrix} a \\ b \\ c \end{pmatrix} \right) \quad (2)$$

$$D = \frac{|ax_1 + by_1 + cz_1 + d|}{\sqrt{a^2 + b^2 + c^2}} \quad (3)$$

식 (3)은 \vec{n} , \vec{v} 의 성분으로 나타낼 수 있으며 이를 정리하면 식 (4)와 같다.

$$D = | \vec{n}^T \cdot \vec{v} + d | \quad (4)$$

한 점에서 평면까지의 거리를 폴리곤 메쉬 모델에 적용시켜 보면, 한 점에서 하나의 폴리곤까지의 거리를 뜻하게 된다. 식(4)에 제곱을 적용하여 Garland 알고리즘을 표현하면 식(5)와 같이 나타낼 수 있다.

$$Q(\vec{v}) = Error_{plane}(\vec{v}) = \sum_i D_i^2(\vec{v}) = \sum_i (\vec{n}_i^T \cdot \vec{v} + d)^2 \quad (5)$$

Garland 알고리즘에서는 식 (5)의 Error를 $Q(\vec{v})$ 라고 정의하였으며, 식의 구성 성분들을 \vec{n} , \vec{v} 로 표현하였다. 식 (5)의 거리 값을 벡터 성분으로 표현하면 Fig. 5로 정리할 수 있다.

$$\begin{aligned}
D^2(v) &= (\vec{n}^T \cdot \vec{v} + d)^2 \\
&= (\vec{v}^T \cdot \vec{n} + d)(\vec{n}^T \cdot \vec{v} + d) \\
&= (\vec{v}^T \cdot \vec{n} \vec{n}^T \cdot \vec{v} + 2d \vec{n}^T \cdot \vec{v} + d^2) \\
&= (\vec{v}^T \cdot (\vec{n} \vec{n}^T) \cdot \vec{v} + 2(d \vec{n}^T) \cdot \vec{v} + d^2)
\end{aligned}$$

Fig. 5 Distance from a vertex to polygon

\vec{n}, \vec{v} 벡터로 표현한 식은 \vec{v} 정점에 대한 이차식으로 나타나게 되는데, 여기서 평면 성분인 \vec{n} , d 성분을 계수로 두어 하나의 이차식을 표현할 수 있게 된다. 이 이차식 성분을 다시 행렬식으로 변환하여 표현한 것을 식 (6)을 통해서 확인할 수 있다. 식 (6)을 하나의 이차 행렬로 표현하여 \vec{v} 정점이 가지는 Quadric error metrics(이하 QEM, 식 (7))라고 정의한다.

$$A = \vec{n} \cdot \vec{n}^T = \begin{bmatrix} a^2 & ab & ac \\ ab & b^2 & bc \\ ac & bc & c^2 \end{bmatrix} \quad b = d\vec{n} = \begin{bmatrix} ad \\ bd \\ cd \end{bmatrix} \quad c = d^2 \quad (6)$$

$$QEM = \begin{bmatrix} A & b \\ b^T & c \end{bmatrix} \quad QEM = \begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{bmatrix} \quad (7)$$

2.2.2 Garland 알고리즘의 적용

각 정점에서 계산된 Q값을 이용해 정점으로 이루어진 모서리의 Q값을 계산할 수 있다. 모서리의 Q값이 최소가 되는 정점을 계산하여, 해당 모서리가 삭제되었을 때 연결이 끊어지는 정점들을 재연결하기 위한 새로운 정점을 계산(식 (8))한다. 이때 Q를 v 에 대하여 미분하여 최솟값을 계산할 수 있으며, 식 (9)와 같이 새로운 정점을 도출할 수 있다.

$$Q(\vec{v}) = Q(\vec{v})_{vertex 1} + Q(\vec{v})_{vertex 2} \quad (8)$$

$$\vec{v} = -A^{-1}b \quad (9)$$

계산된 모서리의 잠재적 새로운 정점인 \vec{v} 를 이용하여 모서리가 삭제될 순서를 정할 수 있다. 모서리의 삭제 순서는 \vec{v} 의 오차값에 의해서 결정되며, 두 정점이 이루는 모서리에 대한 선택 비용(식 (10))이 된다. 이를 바탕으로 모서리의 삭제 순서를 정하고 사용자가 입력한 폴리곤의 수가 도달할 때까지 삭제 함수를 반복 실행한다. 전체 순서는 Fig. 6과 같다.

$$cost = \vec{v}^{-T} (Q_1 + Q_2) \vec{v} \quad (10)$$

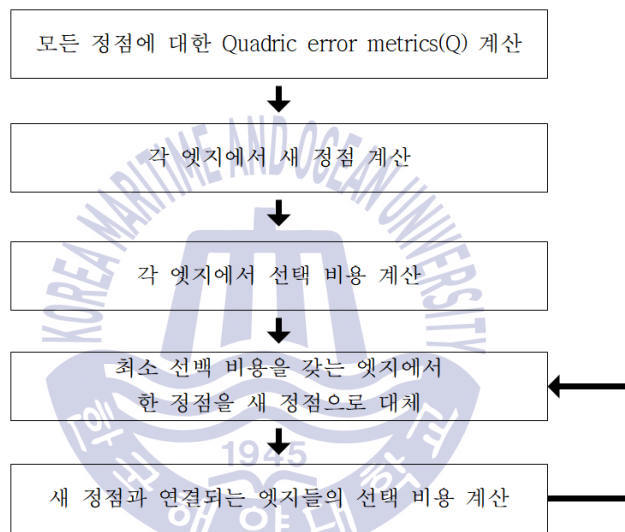


Fig. 6 Process of Garland algorithm

2.3 Metro

Metro(Measuring Error on Simplified Surfaces)는 간략화 된 폴리곤 메쉬 모델의 오차 정도를 측정하기 위한 도구로 제시되었다(Cignoni. et al., 1998). 두 폴리곤들 사이의 오차는 두 폴리곤들이 이루는 단면 사이의 거리(Fig. 7)로 나타낼 수 있다.

$$E(S_1, S_2) = \max_{p \in S_1} e(p, S_2)$$

Fig. 7 Distance between two polygons S_1 and S_2

S_1, S_2 는 두 폴리곤을 나타내고 p 는 S_1 을 구성하는 정점들을 나타낸다. S_1 을 구성하는 정점들과 S_2 사이의 거리 오차(e)들 중 최댓값을 두 폴리곤들의 오차(Fig. 8)로 정의한다.

$$e(p, S) = \min_{p' \in S} d(p, p')$$

Fig. 8 Error of polygon model

여기에서 p 는 임의의 점을 나타내며 p' 는 폴리곤 S 를 이루는 정점들이다. $d(p, p')$ 는 3차원 실수(real) 공간에서 두 점들(p, p') 사이의 유클리드(Euclidean) 거리를 나타낸다. 두 폴리곤들의 오차를 계산 할 때 S_1 과 S_2 의 오차는 교환법칙이 성립하지 않는다.

Metro에서는 두 폴리곤들 사이의 거리를 표현할 때 부호를 적용하였다. 예를 들어 S_1 을 기준으로 하였을 때, Fig. 9와 같이 S_1 의 정점 p_1 에서 가장 가까운 점 $p'(\in S_2)$ 가 S_1 의 밖에 있으면 두 점 사이의 거리(e')는 양수가 되고 그렇지 않으면 음수가 된다.

S_1 의 각 정점들에서의 법선 벡터(normal vector)를 N_p 라 하면 $p(\in S_1)$ 와 p 에서 가장 가까운 점 $p'(\in S_2)$ 사이 거리의 부호는 N_p 와 (p, p') 의 내적에 대한 부호와 같다. 즉, $N_p \cdot (p, p')$ 의 부호와 같다. Metro에서 두 폴리곤들 간의 거리에 대한 부호를 식으로 나타내면 다음과 같다.

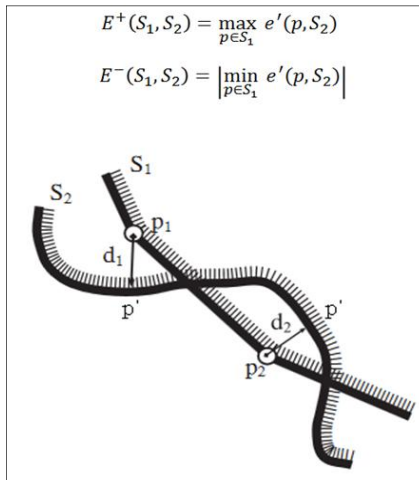


Fig. 9 Evaluate the sign of distance

Metro를 이용하여 두 모델간의 거리를 계산한다(Table 2). 원본 모두 모델이 똑같은 때 이 오차 거리는 0이 된다. 오차 거리가 작을수록 두 모델이 일치하는 정도가 높다고 할 수 있다.

Table 2 Comparison of distances using Metro

Comparison factor	Model 1 -> Model 2	Model 2 -> Model 1
max	0.001095	0.001094
mean	0.000471	0.000463
RMS	0.000502	0.000495
Hausdorff distance	0.001095	

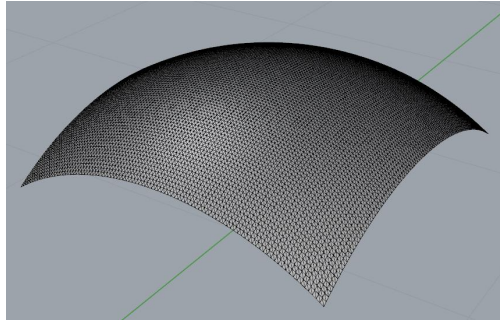


Fig. 10 Model 1 for metro

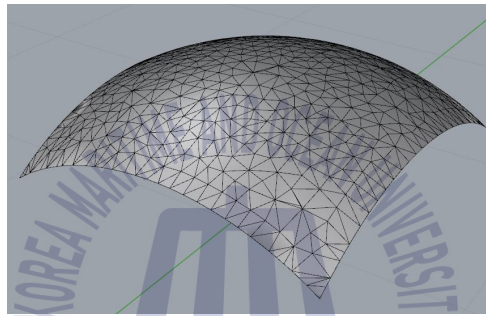
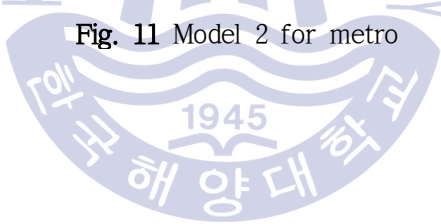


Fig. 11 Model 2 for metro



제 3 장 Garland 알고리즘을 적용 한 해양플랜트 모델의 오류

Fig. 12와 13은 2장에서 설명한 Garland 알고리즘을 모델에 적용한 전후를 비교한 사진이다. Fig. 12, 13은 Garland & Heckbert(1997)에서 기존 알고리즘을 적용한 토끼 모델이고, Fig. 14, 15는 해양플랜트 장비 모델에 적용한 결과이다.



Fig. 12 Original bunny model with 69,451 triangle mesh



Fig. 13 Approximated bunny model with only 1,000 triangle meshes

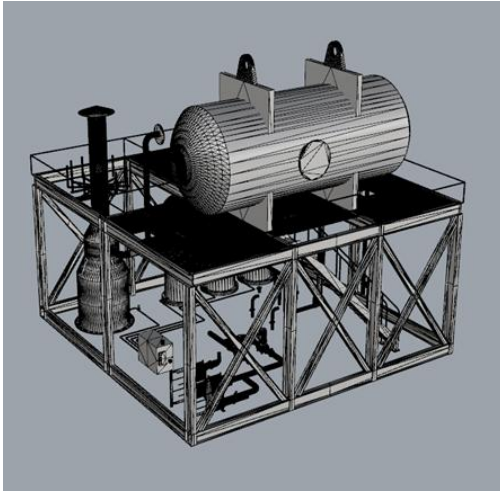


Fig. 14 Original dehydrator model
with 454,202 triangles

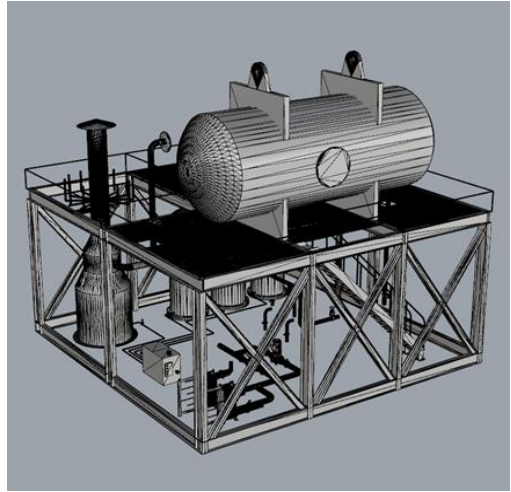


Fig. 15 Approximated dehydrator model
with only 200,000 triangle meshes

Garland 알고리즘을 적용한 예제 모델은 경량화 정도를 조절하여도 그 형태가 보존되는 것을 Fig. 14~17을 통해서 확인할 수 있다.

해양플랜트 모델의 경우도 전체적인 원본 모델(Fig. 14)의 형상을 유지하는 상태에서 폴리곤의 수를 원하는 만큼(Fig. 15) 줄일 수 있었다. 전체적인 모습에는 크게 차이가 없지만, 모델이 경량화 과정에서 발생할 수 있는 오류를 확인해야만 한다. 해양플랜트 모델의 부분 검토를 통해서 오류의 유무를 파악하였다. 경량화 된 모델을 확대하여 세부적으로 확인해 보면 비교적 잘 경량화 된 부분(Fig. 16 ~ 19)이 있지만, 크게 어긋나는 부분(Fig. 20, 21)들을 볼 수 있다.

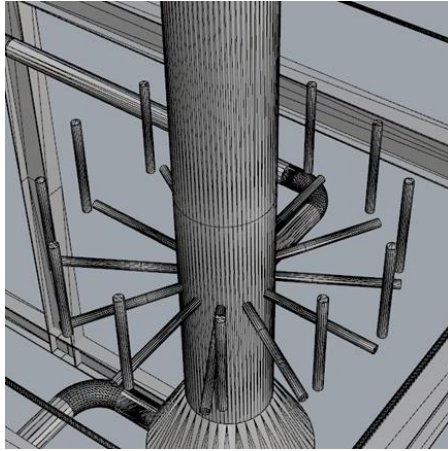


Fig. 16 Original column model with 454,202 triangle meshes

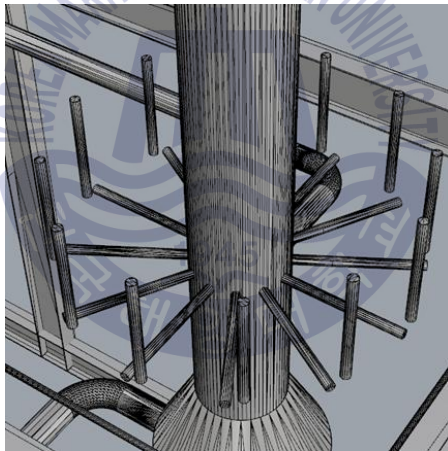


Fig. 17 Approximated column model with only 200,000 triangle meshes

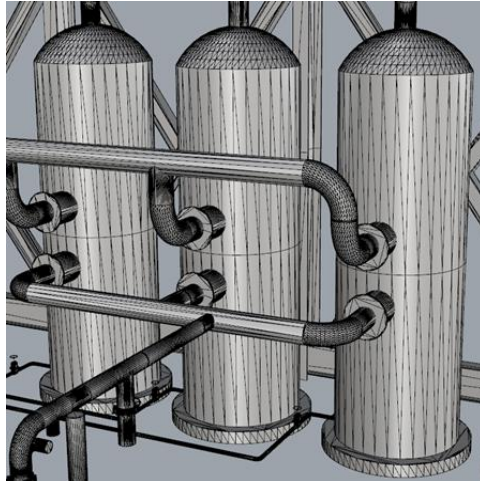


Fig. 18 Original low pressure tank model with 454,202 triangles

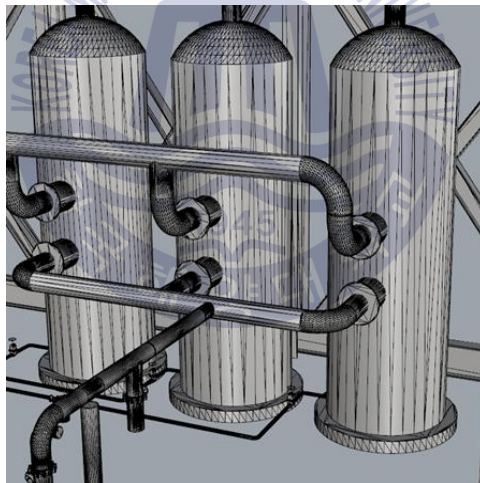


Fig. 19 Approximated low pressure tank model with only 200,000 triangle meshes

3.1 Garland 알고리즘으로 나타난 경량화 모델의 외형적 오류

Fig. 20, 21를 살펴보면, 원본 모델과 경량화 된 모델의 폴리곤 형태가 매우 크게 달라져 있다는 것을 볼 수 있다. 이러한 종류의 경량화의 결과 모델은 명백한 오류로서 그래픽의 현실감을 매우 떨어뜨린다. 뿐만 아니라, 경량화 과정에서 발생한 오류를 설계 단계나 가시화 단계와 같이 다른 과정에서 발생한 오류로 오관할 가능성도 적지 않다. 그렇기 때문에 해당 오류에 대한 원인 분석을 통해 기존 알고리즘에 대한 문제점을 파악하고 해결책을 찾아야 한다.

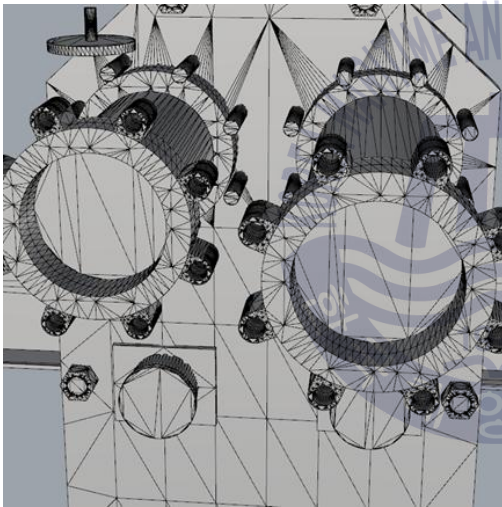


Fig. 20 Original radiator model with 454,202 triangles meshes

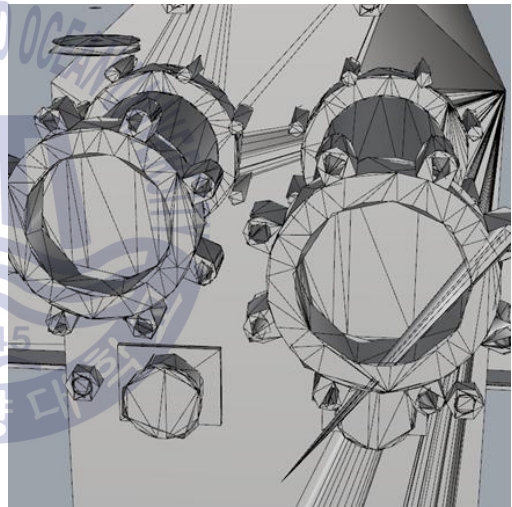


Fig. 21 Approximated radiator model with only 200,000 triangle meshes

Fig. 20과 Fig. 21에 나타나는 경량화 오류에 대한 분석을 진행했다. Garland 알고리즘의 경량화 과정을 살펴보면, 각각의 모서리는 경량화로 삭제될 때, 고유의 새로운 폴리곤 꼭짓점 데이터를 저장하고 있다.

Fig. 22에서 단계 1에 있는 삭제 대상 모서리가 가지고 있는 새로운 폴리곤 꼭짓점은 아래에 위치한 흰색 원의 위치에 있고, 이를 바탕으로 단계 1-3을 진행한다. 새로운 폴리곤 꼭짓점의 위치는 2.2.1절에서 기술한 식 (9)과 같이 계산된다. 식 (9)에서 볼 수 있듯이

모서리가 가지고 있는 QEM(식 (7))의 성분을 이용하여 해당 모서리가 가지는 새로운 폴리곤 꼭짓점을 계산해 낼 수 있다. 새로운 폴리곤 꼭짓점의 계산식(식 (9))에서 사용하는 QEM의 성분은 A 와 b 행렬 성분이다. 두 행렬을 중점으로 모델의 모든 모서리의 QEM 분석을 실시하였다.

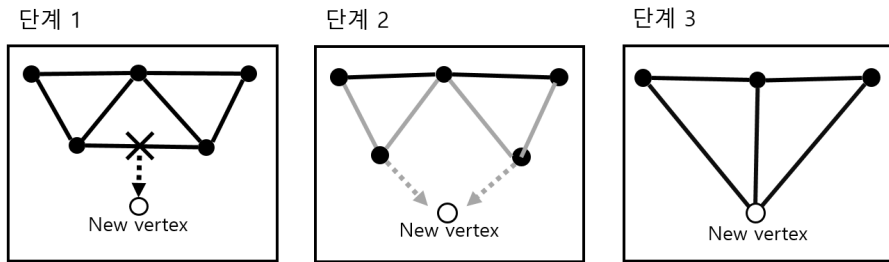


Fig. 22 Simplification process

모서리에 대한 문제 발생의 원인분석 속도 향상을 위하여 문제가 발생하는 부분을 원본 모델에서 분리하였다. 해당 모델(Fig. 23)은 원본 모델에서 오류를 가진 부분을 분리한 모델이며, 5632개의 폴리곤 개수에서 4398개의 폴리곤 개수로 경량화 하였을 때(Fig. 24) 발생하는 케이스로 분석을 진행하였다. Fig. 23의 모든 모서리에 대한 새로운 폴리곤 꼭짓점의 위치를 확인한 결과, Q행렬을 기준으로 모서리를 몇 가지 케이스로 분류할 수 있었다.

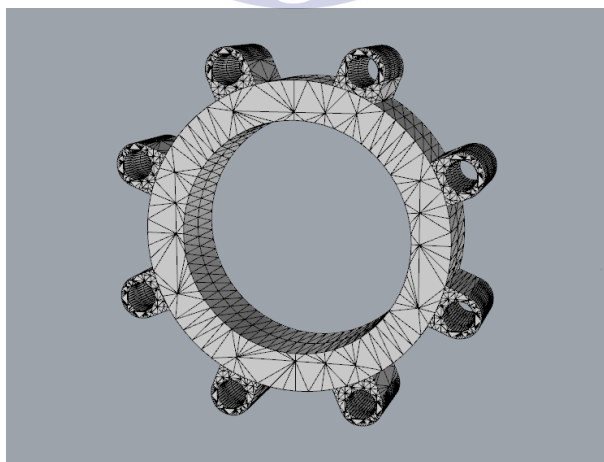


Fig. 23 A part isolated from original model

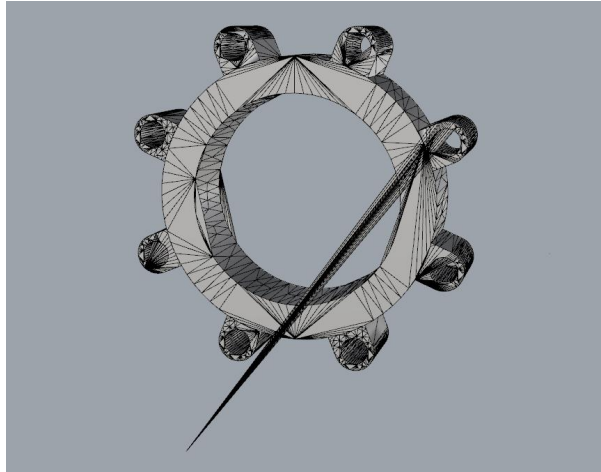


Fig. 24 Case with fatal error, isolated from original model

3.2 정점의 위치에 따른 모서리의 케이스 구분

모서리의 새로운 폴리곤 꼭짓점 위치에 대한 케이스는 크게 3가지로 분류(Fig. 25)할 수 있다. 그 분류의 기준은 식 (9)의 요소에 따라서 달라진다. 첫 번째는 A 와 b 가 바르게 계산 되어 모서리 근처에 새로운 폴리곤 꼭짓점이 위치하는 경우, 두 번째는 식 (9)에서 A 와 b 가 0이 되어 계산이 불가능한 경우, 세 번째는 첫 번째처럼 A 와 b 가 계산되지만 모서리와 상당히 떨어진 곳에 새로운 폴리곤 꼭짓점이 위치한 경우로 나눌 수 있었다.

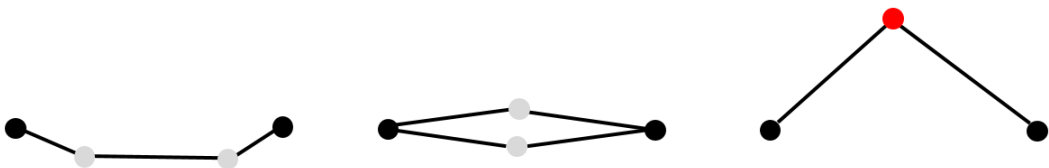


Fig. 25 Case of new vertex for polygon location

첫 번째 케이스는 가장 일반적인 케이스로, Fig. 25의 중앙 그림과 같은 경우로 나타나며 대부분의 모서리가 이에 해당된다. A 와 b 가 올바르게 계산되어 새로운 폴리곤 꼭짓

점이 모서리의 부근에 나타나는 경우이다.

두 번째 케이스는 식 (9)의 두 요소 중 하나가 0이 되어 새로운 폴리곤 꼭짓점이 계산되지 않는 경우이다. 식 (9)에 따르면 QEM의 A 성분의 역행렬이 사용된다. Fig. 26과 같이 x,y,z 좌표계에서 한쪽의 평면에 폴리곤이 위치하게 되면, 이 경우에는 A 행렬을 이루는 \vec{n} 은 $(1, 0, 0)$ 으로 나타난다. YZ 의 평면상에 놓여있기 때문에 법선 벡터의 값은 x 방향으로만 1을 가지게 되고, A 행렬의 성분은 식 (11)처럼 나타나게 된다.

$$A = \vec{n} \cdot \vec{n}^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (11)$$

즉, 이 케이스에서는 A 행렬의 역행렬인 A^{-1} 를 만들어내지 못한다. QEM의 성분을 이용하여 새로운 폴리곤 꼭짓점을 계산할 수 없기 때문에, 경량화 과정에서 오차를 줄이기 위해 모서리의 데이터를 사용한다. Garland & Heckbert(1997)에서는 이러한 경우에 모서리의 중점이나, 모서리를 이루는 두 정점중 하나를 선택하는 차선택을 제시하고 있다.

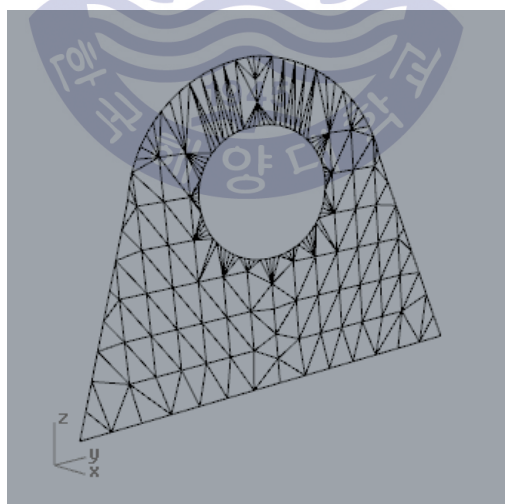


Fig. 26 Polygons in the yz plane

세 번째 케이스는 모서리의 QEM을 이용한 새로운 폴리곤 꼭짓점이 모서리와 매우 먼 거리에 위치한 경우인데, Fig. 21의 우측 그림과 같이 나타난다. 이 경우에 QEM의 A 와 b

를 이용하여 새로운 폴리곤 꼭짓점의 값을 도출한다는 점이 첫 번째 케이스와 유사하다. 그러나, 새로운 폴리곤 꼭짓점의 위치는 첫 번째 케이스와 완전히 다르다. Garland 알고리즘에서는 모델의 외곽을 형성하는 경계 모서리를 살리기 위해 A 와 b 행렬에 가중치(weight)를 더한다. 이 때문에 기존의 Q행렬이 달라지면서 새로운 폴리곤 꼭짓점의 위치에 오류가 발생하는데, 경계 모서리에 대한 자세한 설명은 3.3절과 3.4절에서 다룬다.

Garland 알고리즘에는 세 번째 케이스에 대한 차선책을 제시하지 않고 있는데, 문제는 해양플랜트 모델에서 세 번째 케이스에 해당하는 모서리가 상당수 존재한다는 점이다. 이를 해결하기 위해 세 번째 케이스에 대한 정확한 원인 분석이 필요하다.

3.2.1 깨지기 쉬운 모서리(Breakable edge)

3.2절에서 설명한 세 번째 케이스의 발생 빈도를 파악하기 위해 Fig. 21에 존재하는 모든 모서리의 새로운 폴리곤 꼭짓점의 위치를 출력하였다. Fig. 24 모델의 최대-최소 범위(min-max box) 중 ‘z값을 기준’으로, 모델의 최대-최소 범위를 벗어난 새로운 폴리곤 꼭짓점 데이터를 가지는 모서리를 출력하여 새로운 폴리곤 꼭짓점의 위치를 검토하였다. Fig. 27에 작성된 모서리들은 모두 최대-최소의 범위를 벗어나는 것을 나타낸다. 10만개의 모서리 중에서 약 200여개의 모서리가 검출되었다. 검토의 범위를 ‘최대-최소 범위의 z축을 벗어나는 경우’로 한정하여 검사하였기 때문에, 범위 내에 있지만 크게 어긋나는 새로운 폴리곤 꼭짓점을 가진 모서리의 수는 이보다 더 많을 것으로 판단된다.

이 후부터 이해와 설명을 돕기 위해, 3.2절에서 기술한 세 번째 케이스인 ‘새로운 폴리곤 꼭짓점의 위치가 모서리와 먼 거리를 가지는 경우의 모서리’를 깨지기 쉬운 모서리라고 정의하였다.

번호	X1	Y1	Z1	X2	Y2	Z2	X3	Y3	Z3		
7	440	1933390.000000	8263.546484	39455.277344	441	1933390.000000	8277.564406	39431.652344	1933390.000000	8174.950195	39335.976563
8	442	1933390.000000	8289.754889	39438.406250	445	1933390.000000	8267.828125	39460.730469	1933390.000000	8189.832031	39333.476563
9	873	1933334.000000	8237.980469	39609.511719	881	1933334.000000	8237.557617	39605.066406	1933334.000000	65536.000000	32768.000000
10	873	1933334.000000	8237.980469	39609.511719	882	1933334.000000	8234.657227	39606.019531	1933334.000000	65536.000000	32768.000000
11	874	1933334.000000	8233.606445	39609.511719	882	1933334.000000	8234.657227	39606.019531	1933334.000000	65536.000000	32768.000000
12	881	1933334.000000	8237.557617	39605.066406	882	1933334.000000	8234.657227	39606.019531	1933334.000000	65536.000000	32768.000000
13	992	1933390.000000	8233.606445	39609.511719	990	1933390.000000	8234.657227	39606.019531	1933390.000000	65536.000000	32768.000000
14	993	1933390.000000	8237.980469	39609.511719	989	1933390.000000	8237.557617	39605.066406	1933390.000000	65536.000000	32768.000000
15	993	1933390.000000	8237.980469	39609.511719	990	1933390.000000	8234.657227	39606.019531	1933390.000000	65536.000000	32768.000000
16	999	1933390.000000	8237.557617	39605.066406	990	1933390.000000	8234.657227	39606.019531	1933390.000000	65536.000000	32768.000000
17	1238	1933390.000000	8237.557617	39605.066406	1241	1933390.000000	8234.657227	39606.019531	1933390.000000	65536.000000	16384.000000
18	1239	1933334.000000	8237.557617	39605.066406	1240	1933334.000000	8234.657227	39606.019531	1933334.000000	65536.000000	16384.000000
19	1357	1933334.000000	8311.381836	39540.558594	1378	1933334.000000	8316.955078	39536.730469	1933334.000000	-0.000000	16384.000000
20	1359	1933334.000000	8306.934570	39536.183594	1368	1933334.000000	8303.309570	39537.656250	1933334.000000	16384.000000	-0.000000
21	1359	1933334.000000	8306.934570	39536.183594	1366	1933334.000000	8308.043945	39533.039063	1933334.000000	-0.000000	16384.000000
22	1360	1933334.000000	8306.934570	39540.558594	1368	1933334.000000	8303.309570	39537.656250	1933334.000000	16384.000000	-0.000000
23	1364	1933334.000000	8298.040039	39538.058594	1418	1933334.000000	8296.563477	39560.757813	1933334.000000	-0.000000	16384.000000
24	1374	1933334.000000	8320.276367	39540.558594	1378	1933334.000000	8316.955078	39536.730469	1933334.000000	-65536.000000	32768.000000
25	1423	1933334.000000	8306.934570	39562.429688	1431	1933334.000000	8305.474609	39564.449219	1933334.000000	-65536.000000	16384.000000
26	1465	1933390.000000	8311.381836	39540.558594	1486	1933390.000000	8316.955078	39536.730469	1933390.000000	-0.000000	16384.000000
27	1466	1933390.000000	8306.934570	39540.558594	1476	1933390.000000	8303.309570	39537.656250	1933390.000000	16384.000000	-0.000000
28	1467	1933390.000000	8306.934570	39536.183594	1476	1933390.000000	8303.309570	39537.656250	1933390.000000	16384.000000	-0.000000

Fig. 27 Z values out of the min-max box

3.3 경계 모서리의 역할과 구현

3.2절에서 언급한 세 번째 케이스의 경우 경계 모서리를 유지하기 위해 가중치를 부여한다고 기술하였다. 경계 모서리는 모델에 경량화를 진행하여도 유지하여야 하는 최소한의 형태를 지속시켜 주는 역할을 하는 모서리이다. 경량화는 최대한 원본과의 오차가 적은 형태를 유지하여야 하는데, 가장 크게 유지 시켜야 하는 부분은 모델의 가장 바깥 범위, 즉 경계 모서리이다.

CAD 프로그램에서 객체를 생성하면 Fig. 28과 같이 객체를 이루는 단위가 분할된다. 여기서 각각의 단위의 최외각을 이루는 모서리를 경계 모서리로 정의한다. 이를 프로그램 상으로 정의하기 위해 한 단계 검증 과정을 거친다.

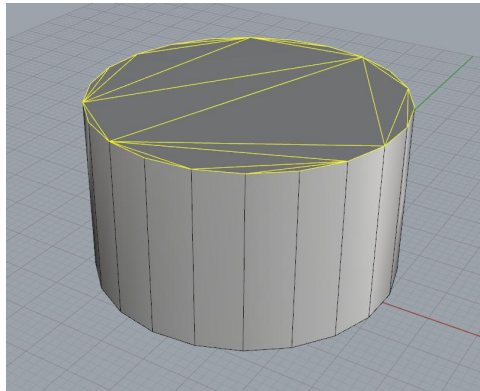


Fig. 28 A 3D model

모서리의 QEM을 생성하는 과정을 살펴보면 Fig. 29와 같은 순서도를 따르게 된다. 모서리를 이루고 있는 정점의 QEM을 통해 해당 모서리의 Q행렬을 계산하게 된다. 그 후에 해당 모서리가 이루는 폴리곤의 개수를 통해서 경계 모서리를 판단한다.

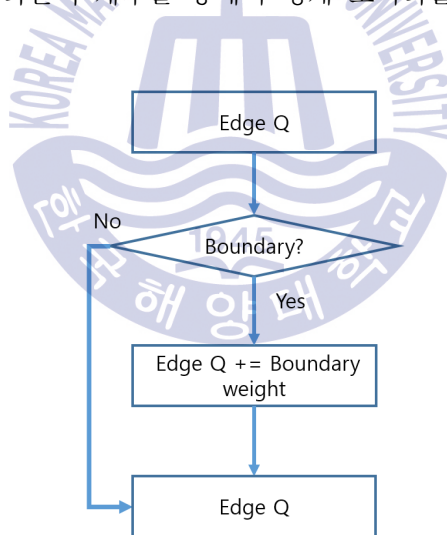


Fig. 29 Q matrix calculation for edge

Fig. 30의 양쪽 그림을 보면, 검은색 모서리가 각각 포함하고 있는 평면의 개수가 다르다. 왼쪽 그림의 경우 평면이 객체의 분할된 단위에서 가장 바깥쪽에 위치한 평면에 해당 되고, 오른쪽 그림의 경우 단위 내부에 위치하는 평면들의 연결 상태를 나타낸다. 왼쪽 그림과 같이 하나의 모서리가 포함하고 있는 평면의 수가 하나일 때, 그 모서리를 경계

모서리라고 판단한다.

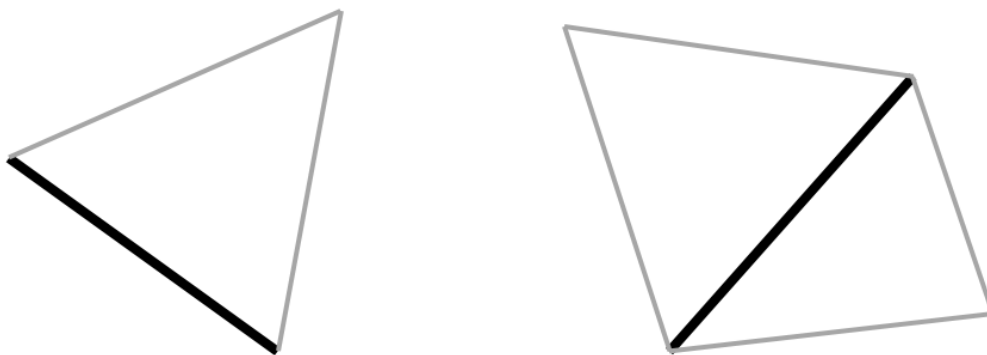


Fig. 30 Number of polygon connected on edge

그 후에 경계 모서리라고 판단된 모서리에 한해서 가중치를 부여하게 된다. 이 가중치 값에 의해서 모서리의 QEM은 초기 정점만으로 계산한 QEM과 다른 값을 가지게 된다.

가중치가 더해진 경계 모서리의 QEM은 자연스럽게 기존 값과 비교하여 매우 커지게 된다. 이로 인해서 작은 QEM을 가진 모서리부터 삭제한다는 Garland 알고리즘의 원칙에 따라, 경계 모서리는 전체 모서리의 삭제 순서에서 뒤로 밀려나게 된다. 이를 통해 경량화를 진행하는 동안 경계 모서리를 최대한 고려하지 않을 수 있게 되며, 모델은 최외각 형태를 유지할 수 있게 된다.

3.4 가중치가 모서리에 끼치는 영향과 범위

Garland 알고리즘에서는 경계 모서리에 가중치를 부여하기 위해 모서리를 이루고 있는 정점에 가중치를 더하는 방법을 사용하고 있다. 정점의 QEM을 이용하여 모서리의 QEM을 생성하기 때문이다. 그러나 경계 모서리에 특수성을 부여하기 위해 정점에 가중치를 더함으로 인해, 경계 모서리 외에 다른 인접한 모서리에도 영향을 끼칠 수 있게 된다.

Fig. 31을 살펴보면 짙은 검은색 경계 모서리에 가중치를 부여하기 위해 이를 이루고

있는 양쪽 질은 검은색 정점에 가중치를 부여한다. 질은 검은색 정점으로 이루어진 경계 모서리의 QEM은 식 (12)로 나타나고, 질은 검은색 정점과 하얀색 정점이 이루는 모서리의 QEM은 식 (13)으로 나타난다. 질은 검은색 정점의 QEM에는 경계 모서리를 유지하기 위한 가중치가 포함되어 있다. 이로 인해 검은색 정점과 하얀색 정점이 이루는 모서리는, 경계 모서리가 아니지만 질은 검은색 정점이 가지고 있는 가중치로 인해 부분적으로 가중치를 가지게 된다. 이러한 방법을 사용하기 때문에 경계 모서리와 연결된 인근 모서리들도 기존의 QEM과 다른 값을 가지게 된다.

$$Q_{black_edge} = Q_{left_black} + Q_{right_black} \quad (12)$$

$$Q_{white_edge} = Q_{black_vertex} + Q_{white_vertex} \quad (13)$$

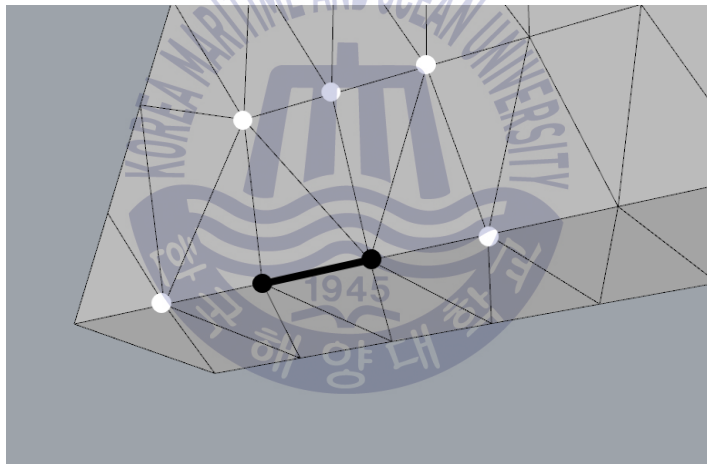


Fig. 31 Weight affect on other edge connected to the boundary edge

이러한 모서리들은 달라진 QEM으로 인해 새로운 폴리곤 꼭짓점의 위치가 달라질 가능성을 지니게 되고, 본 논문의 3.2.1절에서 제시한 깨지기 쉬운 모서리가 될 가능성이 높아지게 된다. 깨지기 쉬운 모서리의 경량화로 인해 발생하는 모델의 일그러짐은 앞서 3.1절에서 언급한 외형적 오류의 문제에서 끝나지 않는다. 깨지기 쉬운 모서리의 경량화로 인해 발생한 현상인 Fig. 24를 살펴보면, 크게 어긋난 한 점으로 인해 많은 모서리들이 연결되어 폴리곤들이 형성된 것을 볼 수 있다. 기존의 QEM 값으로 짜여진 모서리의 삭제 순

서들이, 이 현상으로 생성된 모서리들로 인해 완전히 뒤섞여질 수 있다. 기존의 Garland 알고리즘도 경량화 과정에서 삭제 순서가 갱신되지만, 인접한 폴리곤들이 원본 모델의 경향을 유지하면서 지속된다. Fig. 24의 경우는 경향성이 완전히 무너진 상태에서 갱신이 되기 때문에 모서리의 삭제 순서가 매우 큰 악영향을 미치게 된다.

또한 Fig. 24와 같은 현상이 나타나지 않는 지점, 즉 경량화의 수준이 깨지기 쉬운 모서리가 경량화의 순서가 되기 전까지로 한정될 수밖에 없다. 이 한정되는 경량화의 수준은 모델이 가진 특성에 해당하므로 모델 전체의 QEM을 생성하여 분석하기 전에는 파악할 수 없고, 모델에 적용할 수 있는 경량화의 수준이 매우 낮아질 가능성도 높아지게 된다. 이를 예방하기 위해 깨지기 쉬운 모서리에 대해 새로운 폴리곤 꼭짓점을 선정하는 새로운 기준을 제시한다.



제 4 장 도심 보존을 통한 경량화 개선 알고리즘

4.1 깨지기 쉬운 모서리의 허용 영역 설정

4.1.1 새로운 폴리곤 꼭짓점의 허용 영역

깨지기 쉬운 모서리에 대한 새로운 폴리곤 꼭짓점 결정 방법을 정하기에 앞서 깨지기 쉬운 모서리에 대한 명확한 기준이 필요하다. Fig. 25 오른쪽 그림과 같이 정점이 모서리와 먼 거리를 가지고 있을 때, 이 거리에 대한 판별 기준을 어느 정도로 할 것인가를 정해야 한다. 이를 위해 모서리를 기준으로 특정한 공간을 설정하여 새로운 폴리곤 꼭짓점과 모서리의 위치에 대한 관계를 정리해야 한다. 본 연구에서는 원본 모델과의 오차를 줄이기 위하여, 모서리를 단위로 활용할 수 있는 기하학적 공간으로 깨지기 쉬운 모서리의 기준을 제시한다. 설정한 공간 안에 새로운 폴리곤 꼭짓점이 생성되어야 경량화 전과 비교하여 모델의 오차를 최소화 할 수 있을 것이라고 판단하였다.

깨지기 쉬운 모서리의 판별을 위한 공간의 기준을 선정하기 위해 고려한 두 가지 사항은 알고리즘의 속도와 정당성이다. 수많은 모서리에 대해 깨지기 쉬운 모서리의 판단 검토를 해야 하기 때문에 판단 공간 영역을 구현하는 알고리즘은 가벼워야 한다. 그리고 영역 밖으로 벗어난 새로운 폴리곤 꼭짓점이 경량화에 악영향을 미치는 것으로만 이루어져

있는지에 대한 검토가 필요하다.

4.1.2 허용 영역 : 구

깨지기 쉬운 모서리의 새로운 폴리곤 꼭짓점을 판단하는 공간의 구현에 있어서 가장 중요한 것은 속도이다. Garland & Heckbert(1997)에서 사용한 토끼 모델(Fig. 12)의 경우 약 7만개의 폴리곤으로 이루어져 있고 10만개 이상의 모서리를 가지고 있다. 해양플랜트 모델은 이와 비교 할 수 없을 정도로 많은 모서리를 가지고 있다. 그렇기 때문에 판단 공간 영역의 구현 속도는 무시할 수 없는 큰 요소이다.

공간 설정을 위해 모서리를 기준으로 하여 구현할 수 있는 기하학적 공간들을 비교하였다. Fig. 32와 같은 기하학적 공간들의 구현 알고리즘들을 비교하였다. Fig. 32에 나타난 2개의 하얀색 원은 모서리를 이루는 정점을 의미한다. 모서리를 대각으로 가지는 정육면체와 지름으로 가지는 구, 그리고 높이로 가지는 원기둥을 대표적 공간으로 비교하였다.



Fig. 32 Examples of geometric spaces

알고리즘에 대한 검토 후 가장 간단한 영역으로 확인된 구를 선정하였다. 다른 공간 표현에 비해, 단순한 길이비교를 통해 구 영역을 표현할 수 있다. 모서리를 지름으로 가지는 구 영역의 경우, 모서리의 중점과 모서리가 가지는 새로운 폴리곤 꼭짓점의 거리를 비교하는 것으로 간단하게 구현(Fig. 33)할 수 있다.

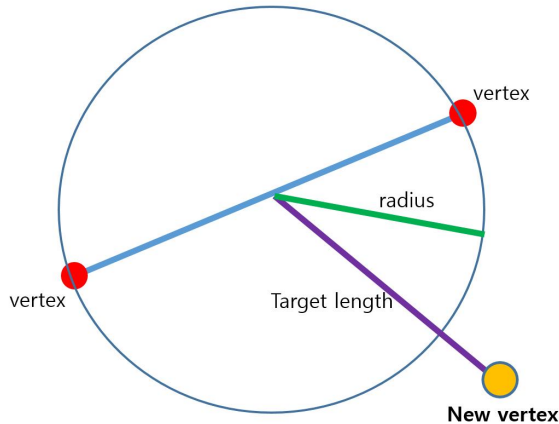


Fig. 33 Allowable space set by a sphere

모서리를 지름으로 하는 구를 구현하기 위해, 모서리의 중점과 모서리의 새로운 폴리곤 꼭짓점이 이루는 거리를 측정하고 이를 목표 반경(Target length)라고 정의한다. 이 목표 반경과 모서리의 길이 절반을 반지름으로 하여 이를 비교한다. 그리고 이 둘의 대소를 비교하여 모서리의 새로운 폴리곤 꼭짓점이 구 영역 안에 위치한지를 판단한다. 그러므로 모든 모서리에 대한 판단이 이루어지기 위해 가벼운 알고리즘을 가져야 한다는 조건에 부합하는 공간 영역이라고 결론지을 수 있다.

4.1.3 정당성 검토

설정된 허용 영역을 벗어난 새로운 폴리곤 꼭짓점을 가지는 모서리들이 모두 깨지기 쉬운 모서리라고 할 수 있는지를 검토하였다. 모서리의 허용 영역은 경량화에 큰 걸림돌이 되는 깨지기 쉬운 모서리를 예방하기 위해 설정한 영역이다. 그러나 모서리의 새로운 폴리곤 꼭짓점이 허용 영역 밖에 위치하면서도, 직관적으로 보았을 때 경량화에 영향을 적게 주는 경우가 존재할 수 있다.

Fig. 34를 살펴보면 모서리를 기준으로 설정한 구 영역을 볼 수 있다. 이 영역을 기준으로 정점의 위치에 따라 3가지 정도로 케이스를 구분할 수 있다. 케이스 3과 같이 깨지기 쉬운 모서리의 새로운 폴리곤 꼭짓점인 경우와 케이스 1과 같이 허용 영역 안에 위치한 경우, 그리고 마지막으로 케이스 2처럼 허용 영역 밖에 있지만 모서리와 거리에 멀지 않아서 경량화에 큰 영향을 끼치지 않는 경우가 있다.

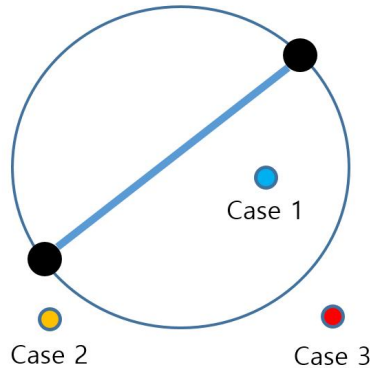


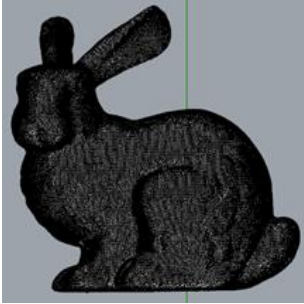
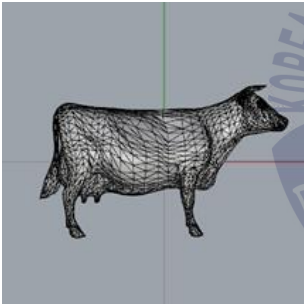
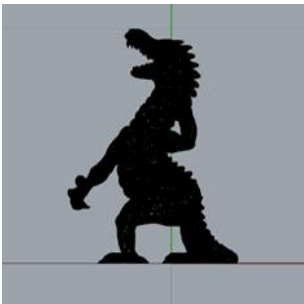
Fig. 34 Location analysis of new vertices

기존의 알고리즘의 문제 해결을 위해 설정한 허용 영역이므로, 기존 알고리즘으로 테스트 했던 모델을 통해 모서리의 새로운 폴리곤 꼭짓점위치들을 출력하여 하나의 모델이 가지는 케이스들의 비율을 분석하였다. 한 모델에서 케이스 2와 같은 점의 경우가 많이 나타난다면, 경량화에도 오차를 크게 가지지 않지만 4.1.2절에서 설정한 허용 영역에 의해 깨지기 쉬운 모서리로 정의되는 경우가 발생할 수 있다. 이는 허용 영역 설정의 동기에 대한 모순으로 설정될 수 있으므로, Fig. 34의 케이스별 비율에 대한 검증이 필요하다.

Garland & Heckbert(1997)에서 실행한 예제 모델들을 통해 해당 모델에서 Fig. 34의 케이스 2가 얼마나 발생하는지를 살펴보았다. 여기서 테스트 한 모델들은 경계 모서리가 존재하지 않기 때문에 가중치가 없는 모델, 깨지기 쉬운 모서리가 존재하지 않는 모델이다. 여기서 발생하는 모서리의 새로운 폴리곤 꼭짓점 중 허용 영역을 벗어나는 것은 케이스 2에 해당하는 경우이다.

케이스 2에 해당하는 모서리의 개수를 파악한 결과는 Table 3과 같다. 각각의 모델에서 영역 밖에서 나타나는 점들의 비율을 살펴보면, 3가지 검토 모델 모두 1%가 되지 않는 아주 작은 수치를 가지고 있는 것을 알 수 있다. 즉 모서리를 지름으로 하는 구 영역을 허용 영역으로 정하였을 때, 이를 벗어나는 위치에 있는 새로운 폴리곤 꼭짓점의 발생 빈도는 이를 무시할 수 있을 만큼 극소수로 나타난다.

Table 3 Comparison of ratios of outside edge

Model	number of total edge	number of edges outside allowable area	Ratio(%)
	104,288	304	0.29
	8,706	68	0.78
	76,175	45	0.06

4.1.2절과 4.1.3절을 통해 선정된 공간 영역인 구를 이용하여 깨지기 쉬운 모서리의 기

준을 선정하였다. 구 영역은 가벼운 알고리즘을 가지고 있고, 기존 경량화 과정에서 이를 벗어나는 정점 생성하는 비율이 매우 낮다. 그러므로 구로 설정한 허용 영역을 벗어난 새로운 폴리곤 꼭짓점을 가지는 모서리는 깨지기 쉬운 모서리라고 정의 할 수 있다.

4.2 깨지기 쉬운 모서리의 새로운 폴리곤 꼭짓점 선정

3장에서 분석한 문제의 원인과 4.1절에서 제시한 허용 영역을 기준으로 하여, 깨지기 쉬운 모서리에 대해 새로운 폴리곤 꼭짓점을 구하기 위한 다른 계산법을 적용해야 한다. 기존의 알고리즘의 경량화 기준은 QEM을 이용한 결과 값이었지만, 경계 모서리로 인해 깨지기 쉬운 모서리가 생기는 것을 확인할 수 있었다. 깨지기 쉬운 모서리는 기존의 알고리즘으로 경량화에 적당한 새로운 폴리곤 꼭짓점을 생성할 수 없으므로, 폴리곤이 가지는 다른 요소를 통한 경량화 기준이 필요하다.

깨지기 쉬운 모서리의 가장 큰 문제점은 새로운 폴리곤 꼭짓점의 위치가 모서리와 먼 거리를 가진다는 것이다. 그렇기 때문에 원본과 비교하여 경량화가 진행된 후에 폴리곤들이 나타내는 형상은 큰 오류를 가지게 된다. 이를 해결하기 위해, 깨지기 쉬운 모서리의 새로운 폴리곤 꼭짓점으로 발생하는 폴리곤의 변화가 원본 모델과 비교하였을 때 가지는 오차가 작아야 한다는 것을 바탕으로 경량화의 기준을 제시한다.

4.2.1 모서리의 경로에 존재하는 점을 이용한 새로운 폴리곤 꼭짓점 선택

원본 모델과의 오차를 작게 만들기 위한 첫 번째 조건은, 원본 모델에 위치하는 객체를 이용한 한 점을 선택하는 것이다. 기존의 알고리즘에서도 이와 유사하게, 4.1절에서 언급한 두 번째 케이스의 경우에 대한 대비책으로 모서리의 경로에 위치하는 한 점을 선택하고 있다. 원본 모델에 존재하는 모서리나 정점 정보를 이용하는 것이 경량화 후에 그 오차를 작게 할 수 있다는 것을 선행 연구를 통해 알 수 있다. 본 연구에서도 Fig. 35와 같이 모서리의 경로에 위치하는 점을 선택함으로써 원본 모델과의 오류를 최소화하였다.

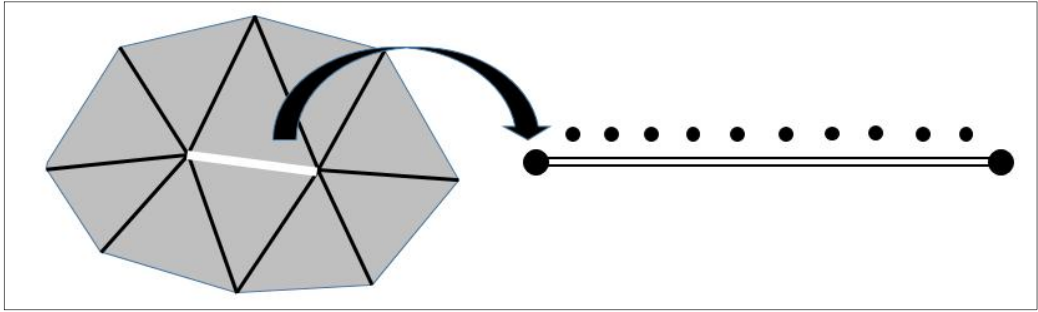


Fig. 35 10 points selected on edge

4.2.2 폴리곤 그룹 설정과 비교

두 번째로 정한 조건은 인접한 다른 폴리곤들에 주는 영향을 최소화 하는 것이다. 4.2.1 절에서 제시한 조건을 만족 시킨 후, 모서리와 연결된 인접 폴리곤들의 변화를 최소화 하였다.

하나의 모서리가 삭제되는 현상은 그 과정에서 모서리를 포함하는 폴리곤들에 외형적으로 가장 큰 영향을 준다. 모서리를 포함하고 있는 폴리곤들은 해당 모서리가 삭제되면 같이 사라지게 된다. 그렇기 때문에 삭제되는 폴리곤과 더불어 원본모델과의 오차를 최대한으로 줄이기 위해서 집중해야 하는 폴리곤은, 모서리의 삭제로 인해 사라지는 폴리곤이 아닌 그 형태가 변하는 폴리곤들이다. 이러한 폴리곤에는, 모서리의 정점과 연결되어 있는 폴리곤들이 해당된다. 정점과 연결되어 있는 폴리곤들은, 모서리의 삭제로 인해 기존의 정점과 연결되어 있던 모서리들이 이동되면서 그 모양이 바뀌게 된다.

경량화의 과정에서 삭제되거나 변형되는 폴리곤들의 성질을 최소화하여야, 경량화의 전체적인 흐름과 원본 모델과의 차이를 최소화 할 수 있다. 이를 위하여 깨지기 쉬운 모서리와 연결되어 있는 모든 폴리곤들을 그룹화 하여 그 특성을 비교하였다. 모서리를 직접적으로 포함하는 폴리곤과, 모서리를 이루는 정점을 포함하는 폴리곤들을 그룹화하여 최종적으로 모서리를 완전히 감싸는 하나의 폴리곤 그룹(Fig. 36)을 생성한다.

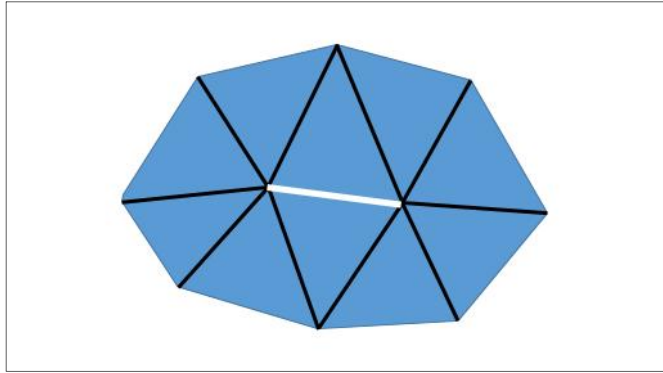


Fig. 36 Original polygon group and breakable edge

하나의 폴리곤 그룹은 모서리의 삭제로 인해 무조건적으로 변하게 되는 폴리곤들의 집합이다. 즉, 이 그룹 내의 폴리곤들이 최소한의 변화를 가져야 원본과의 오차를 최소한으로 가지게 된다. 이를 위하여 폴리곤 그룹 내의 폴리곤들의 기하학적 요소의 변화량을 이용하였다.

폴리곤 그룹이 경량화 전후에 가지는 면적과 도심의 변화에 주목했다. Fig. 35처럼 모서리의 경로에 있는 한 점을 선택하여(Fig. 37), Fig. 38과 같이 기존의 모서리의 정점과 연결되어 있던 모서리들이 그 점으로 연결된다고 가정하였다. 그 후 원본 모델의 폴리곤들이 가지는 면적의 변화와 도심의 변화를 비교하였다. 모서리의 경로에서 새로운 폴리곤 꼭짓점의 위치를 결정하기 위해, 면적과 도심의 변화량이 최소한으로 변화되는 지점을 선정하였다. 본 연구에서는 모서리의 폴리곤 그룹이 형성하는 가장 바깥에 위치한 모서리들을 최외각 모서리로 설정(Fig. 37)하였다.

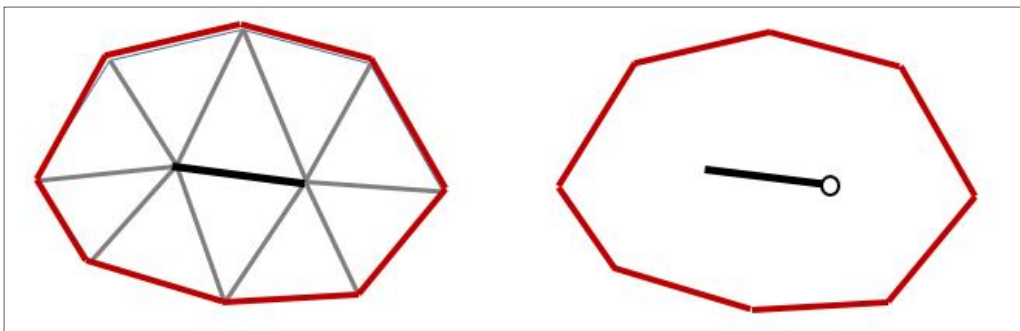


Fig. 37 Original polygon group and outermost edge

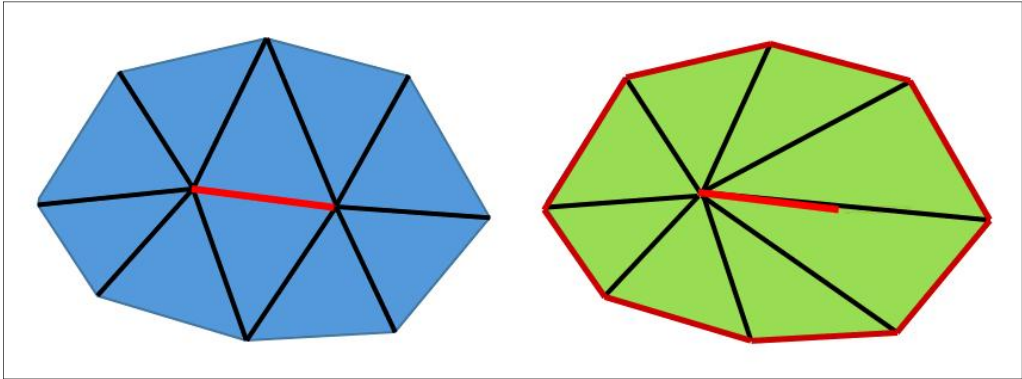


Fig. 38 Original model and one of case points on edge

그 후 최외각 모서리들의 정점과 기존 깨지기 쉬운 모서리의 정점에 연결되어 있는 모서리의 끝점을 연결하여 새로운 모서리를 생성하였다.

모서리를 10등분한 점들 중 한 점에, 기존 정점과 연결되어 있던 모서리를 연결하여 경량화를 하였다고 가정하고, 이때 폴리곤의 변화를 비교하였다. 원본 폴리곤들이 가지는 면적·도심의 정보와, 모서리를 10등분한 점에 연결하였을 때 생성되는 폴리곤의 정보를 비교하는 과정을 설정했다. 이 과정을 하나의 케이스로 설정하여, 10등분 한 점의 각 케이스마다 산출되는 면적과 도심의 변화를 비교하였다.

모서리를 10등분한 이유는, 그 이상을 분할하여 폴리곤 그룹의 변화를 비교하였을 때와 큰 차이를 보이지 않았기 때문이다. 물론 더 많은 점을 분할하였을 때 좀 더 정확한 데이터를 얻을 수 있지만, 전체 알고리즘의 속도를 많이 떨어트릴 뿐만 아니라 10등분을 했을 때와 경량화 결과에서 큰 차이를 보이지 않았다는 점을 고려하였다.

4.2.2.1 면적 비교

Fig. 39와 40은 3D 환경에서의 예제를 생성하여 4.2.2절에서의 과정을 보여주는 모델이다. Fig. 39와 같은 폴리곤 그룹의 전체 면적을 통해, Fig. 36과 같은 10개의 케이스를 생성하였다. Fig. 40에서 하얀색으로 표시되어 있는 모서리는 원본 모델에서의 깨지기 쉬운 모서리이다. Fig. 39에서 나타나는 10개 폴리곤 면적의 합이 깨지기 쉬운 모서리의 삭제 후 재 생성되는 폴리곤들의 전체 면적 합을 비교한다.

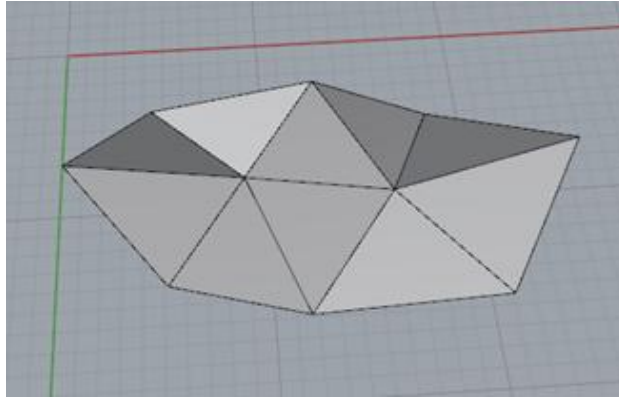


Fig. 39 Original polygon group in 3D

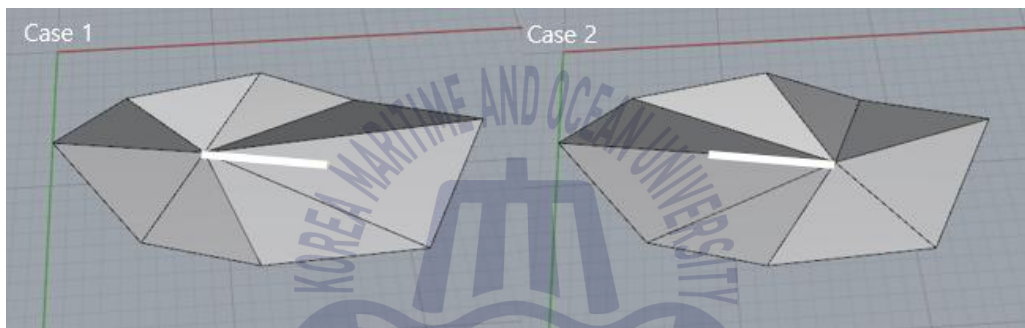


Fig. 40 Cases of simplified model connect to a point in the path of a breakable edge

폴리곤 그룹의 최외각 모서리의 영역 내에서 일어나는 전체 면적의 비교를 통해, 기존의 폴리곤들이 가지고 있는 특성의 변화를 최소화 하였다.

4.2.2.2 도심 비교

4.2.2절에서 생성한 10개의 케이스를 4.2.2.1절에서는 폴리곤들의 면적을 이용하여 비교했다. 폴리곤들의 면적 비교는 기존의 성질을 최대한 유지할 수 있지만, 폴리곤 그룹의 면적이 비교적 일정해야 한다는 조건이 있다. 폴리곤 그룹 내 개별 요소들의 변화의 총합을 이용하기 때문에, 비교를 위한 변수는 사실상 폴리곤 그룹의 면적이라고 할 수 있다. 그런 이유에서 폴리곤 그룹의 요소들이 불규칙할 때는 다소 불안한 결과 값을 도출하였다.

불규칙한 폴리곤 그룹에 각 폴리곤들의 성질을 유지하기 위해 도심을 이용하여 새로운 폴리곤 꼭짓점을 선정하였다. 폴리곤들의 전체 면적 비교가 아닌, 폴리곤 그룹의 개별 객체들이 얼마나 움직이는지를 비교하기 위해 각각의 폴리곤들이 변화하는 도심의 변화(Fig. 41)에 초점을 맞추었다.

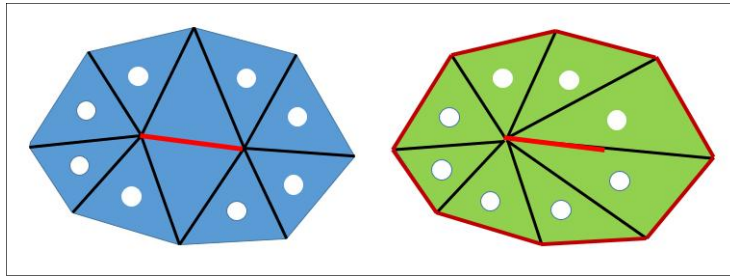


Fig. 41 Translation of centroid

4.2.3 도심이 가진 기하학적 특성

4.2.2절에서 설정한 폴리곤 그룹을 면적과 도심을 기준으로 비교하여 보면 Fig. 42, 43와 같이 큰 차이 없이 유사하게 경량화 하는 것(Fig. 42, 43)을 확인할 수 있다.

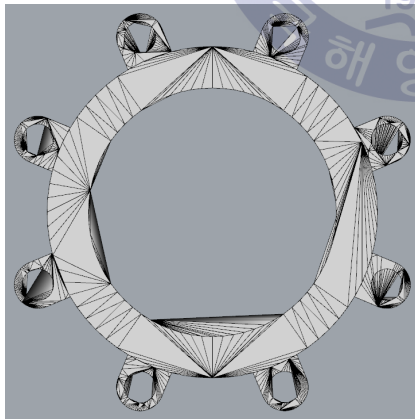


Fig. 42 Result of area

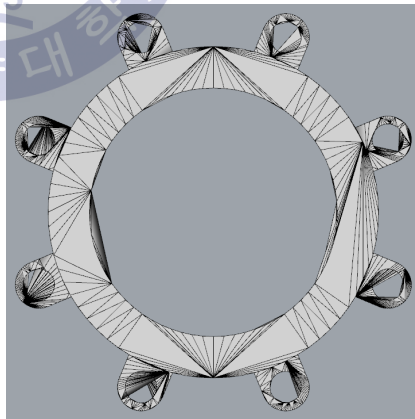


Fig. 43 Result of centroid

면적은 벡터(vector) 데이터인 정점들을 이용하여 만들어진 스칼라(scalar) 데이터이고, 도심은 폴리곤이 가지고 있는 정점을 활용하여 나타내는 벡터 데이터이다. 스칼라 형태

의 면적 데이터는 공간상에 표현할 수 없지만 도심 데이터는 공간상에 표현할 수 있다. 경량화는 공간에 존재하는 벡터 데이터로 이루어진 폴리곤의 변화를 최소화 하여야 하는 과정이다. 스칼라 데이터인 면적의 변화보다 벡터인 도심의 변화를 최소화 하는 것이 폴리곤 데이터를 효과적으로 유지할 수 있는 방법이다.

이를 뒷받침하기 위해 면적과 도심을 이용한 폴리곤 그룹의 변화를 각각 도출한 후, 원본 모델과의 일치하는 정도를 비교하였다. 두 물체의 정점과 폴리곤 사이의 거리오차를 계산하는 방법으로 Metro를 활용하였다(Cignoni et al. 1998). Metro에서 제공하는 최대 거리 및 평균 오차 값을 이용하여 경량화 모델의 일치도로 활용하였다.

Fig. 44와 47은 각각 면적과 도심을 이용한 경량화를 Metro를 이용하여 원본과의 오차를 계산한 결과이다. Table 4을 통해서 거리 결과값들의 차이를 보면 도심을 이용한 오차가 면적의 오차보다 더 작다는 것을 확인할 수 있다.

```

Forward distance (M1 -> M2):
target # samples      : 56320
target # samples/area : 0.312792
Vertex sampling
Edge sampling

distances:
  max : 0.080134 (0.000209 wrt bounding box diagonal)
  mean : 0.010214
  RMS  : 0.015426
# vertex samples      4668
# edge samples        67264
# area samples         0
# total samples       71932
# samples per area unit: 0.399499

Backward distance (M2 -> M1):
target # samples      : 56320
target # samples/area : 0.300628
Vertex sampling
Edge sampling

distances:
  max : 12.262821 (0.082022 wrt bounding box diagonal)
  mean : 0.097693
  RMS  : 0.644009
# vertex samples       3702
# edge samples         70770
# area samples         0
# total samples       74472
# samples per area unit: 0.397521

Hausdorff distance: 12.262821 (0.082022 wrt bounding box diagonal)
Computation time   : 198 ms
# samples/second   : 739414.106294
  
```

Fig. 44 Error between original & simplified model after simplified using area variation

```

Forward distance (M1 -> M2):
target # samples      : 56320
target # samples/area : 0.312792
Vertex sampling
Edge sampling

distances:
  max : 0.078937 (0.000206 wrt bounding box diagonal)
  mean : 0.010371
  RMS  : 0.015568
# vertex samples      4668
# edge samples        67264
# area samples         0
# total samples       71932
# samples per area unit: 0.399499

Backward distance (M2 -> M1):
target # samples      : 56320
target # samples/area : 0.301906
Vertex sampling
Edge sampling

distances:
  max : 9.942577 (0.025963 wrt bounding box diagonal)
  mean : 0.088937
  RMS  : 0.562130
# vertex samples      3716
# edge samples        71867
# area samples         0
# total samples       75583
# samples per area unit: 0.405167

Hausdorff distance: 9.942577 (0.025963 wrt bounding box diagonal)
Computation time   : 224 ms
# samples/second   : 658549.075863

```

Fig. 45 Error between original & simplified model after simplified using centroid variation

Table 4 Comparison of distance of both method using metro

Comparison factor	Area		Centroid	
	Original-> simplified	simplified -> Original	Original-> simplified	simplified -> Original
max	0.080134	12.262821	0.078937	9.942577
mean	0.010214	0.097693	0.010371	0.088937
RMS	0.015426	0.644009	0.015568	0.562130
Hausdorff distance	12.262821		9.942577	

제 5 장 결론

5.1 결과

개발한 알고리즘의 순서도는 Fig. 46과 같으며, 이 과정을 기존의 알고리즘 순서에서 모서리의 새로운 정점을 계산하는 단계에 추가하였다. 이렇게 개선된 알고리즘을 통해서 오류가 발생하였던 해양플랜트 모델을 경량화 하여 오류 발생 여부를 확인하였다.

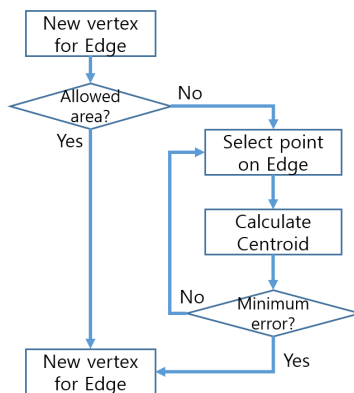


Fig. 46 Algorithm for selection of new vertex on edge

새로이 제시하는 경량화 기준을 이용한 알고리즘은, 깨지기 쉬운 모서리에 대한 새로운 폴리곤 꼭짓점의 선정 방법이다. 모서리의 경로에서 새로운 폴리곤 꼭짓점의 후보군을 뽑아내고, 각 폴리곤들이 가지는 도심의 최소 오차점을 선택한다. 그렇기 때문에 기존 폴리곤들의 성질을 최대한 유지할 수 있고, 그와 동시에 최소한의 오차를 가진 경량화를 진행할 수 있었다.

기존의 알고리즘에서 발생한 문제가 Fig. 42, 45을 통해 해결 된 것을 확인할 수 있다. 또한 Fig. 47을 통해서 추가적으로 확인 할 수 있는 것이 있다. 경량화를 단계를 더 끌어 올릴 수 있다는 점이다. Fig. 47의 왼쪽 사진은 기존 알고리즘에서 12%만큼 폴리곤을 경량화 한 사진이다. 위와 같은 오류가 발생하기 때문에 그 이상의 경량화가 불가능하다. 개선된 알고리즘의 적용을 통해서 오류에 대한 대안을 적용하여, 약 65%의 폴리곤을 경량화 한 사진(Fig. 47 오른쪽)을 확인 할 수 있다.

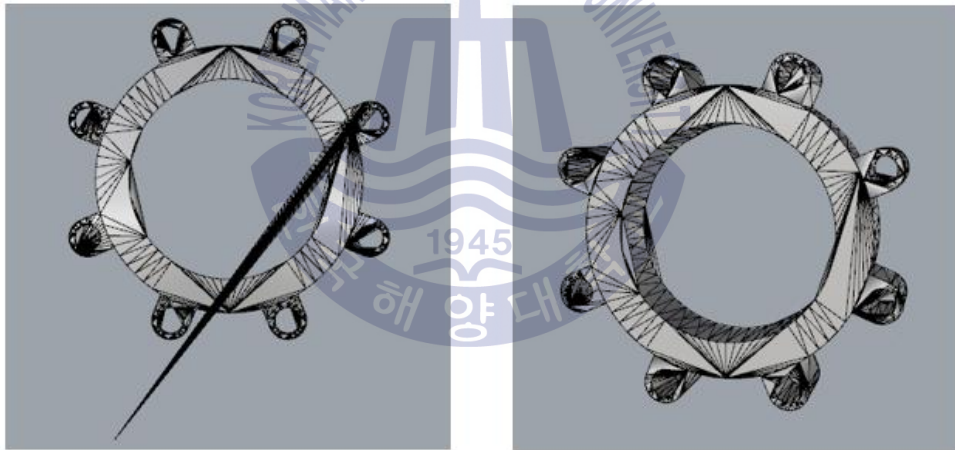


Fig. 47 Further simplification at higher levels

7.2 향후 과제

깨지기 쉬운 모서리를 정의하는 기준으로 본 연구에서는 모서리를 지름으로 하는 구를 제시하였다. 대부분의 폴리곤 메쉬 모델은 비교적 비슷한 크기의 폴리곤들을 가지고 있어, 허용 영역이 효과적으로 깨지기 쉬운 모서리를 판별한다. 하지만 해양플랜트 모델의

경우 수많은 파이프 모델들이 존재하고, 파이프는 그 길이만큼의 삼각형을 기본 메쉬 단위로 가지기 때문에 허용 영역이 매우 커질 수 있다. 파이프는 비교적 경량화의 순서에서 뒤에 자리 잡지만 이 또한 언제든지 깨지기 쉬운 모서리가 될 가능성이 높다. 이를 위하여 깨지기 쉬운 모서리에 대한 좀 더 명확하고 근거 있는 영역의 기준에 대하여 연구할 필요가 있다.



References

- Botsch M., Kobbelt L., Pauly M., Alliez P. & Levy B., 2010. *Polygon Mesh Processing*. A K Peters, Ltd.
- Cignoni P., Rocchini C. & Scopigno R., 1996. Metro: measuring error on simplified surfaces. *Computer Graphics Forum*, 17(2), pp.167-174.
- Garland M. & Heckbert P. S., 1997. Surface simplification using quadric error metrics. *SIGGRAPH '97 Proceedings of the 24th annual conference on Computer graphics and interactive techniques.*, pp.209-216.
- Hoppe H., 1996. Progressive meshes. *SIGGRAPH '96 Proceedings of the 23rd annual conference on Computer graphics and interactive techniques.* pp.99-108.
- Piegl L. & Tiller W., 1997. *The NURBS Book*. Springer.
- William J.S., Jonathan A.Z. & William E.L., 1992. Decimation of triangle meshes. *SIGGRAPH '92 Proceedings of the 19th annual conference on Computer graphics and interactive techniques.* pp.65-70.

권순조, 김병철, 박선아, 문두환, 한순홍, 2015. 플랜트 유닛 수준의 대용량 3D CAD 조립체의 간략화 방안. *한국CAD/CAM학회 2015 동계학술대회 논문집*, 2015.2, pp.71-74

김병철, 권순조, 박선아, 문두환, 한순홍, 2014. 조선해양 기자재 3D CAD 단품 데이터 간략화 시스템 개발. *한국CAD/CAM학회 논문집*, pp.167-176.

천상욱, 박광필, 서홍원, 2011. 3D 기반의 협업을 위한 모델 경량화 기술 및 Simulation 응용. *대한조선학회지* 48(3), pp.55-60.

