

3D모델의 생애주기 활용을 위한 객체 분할 알고리즘의 정확성 비교 연구*

A Comparative Study on the Accuracy of Object Segmentation Algorithms for Lifecycle Utilization of 3D Models

주현철** · 강인석***

Hyun-Chul Joo** · Leen-Seok Kang***

Abstract

While the use of 3D models during the design phase is increasing, their usefulness throughout the lifecycle is diminished due to the differences in the Level of Detail (LoD) compared to the 3D models used in the construction and maintenance phases. To utilize 3D models created during the design phase for construction schedule management and maintenance damage information management, segmentation of the 3D model is necessary. This study compares vertex data-based methods, BSP and K-D tree algorithms for 3D model segmentation. The accuracy of the each segmentation method is verified and an appropriate segmentation methodology is proposed. The research targeted a bridge pier model and extracted cylindrical, rectangular column, and coping objects. These were then segmented into appropriate divisions using each methodology, and the accuracy of the segmentation methods was evaluated by comparing volumes before and after segmentation. As a result, the vertex data-based segmentation method showed the smallest error in the cylindrical sections and the lower coping, and the next smallest error in the rectangular columns and upper coping, confirming its suitability as a segmentation methodology for 3D models.

Keywords: Vertex-Data, BSP, K-D Tree, 3D Model Segmentation

1. 서론

일반적으로 교량이나 건축구조물의 손상 여부 검사는 일차적으로 육안 검사를 시행하고 있으며, 이를 통해 개략적인 손상 발생 위치와 손상 규모를 판단할 수는 있으나 점검할 때마다 손상 발생과 위치 정의에 있어 일관성이 부족해질 수 있다(양다현 외,

2023). 최근에는 노후 구조물의 손상정보 관리를 위해 BIM(Building Information Modeling) 기반의 3D모델을 활용한 유지관리 기술이 점차 보편화되고 있다(이상호 외, 2023; 최용규 외, 2024). 3D모델을 활용한 유지관리기술은 설계단계에 생성한 모델을 구조물의 생애주기 동안 재활용할 수 있고, 균열정보를 3D기반의 시각적 정보로 관리할 수 있는 점에

*본 연구는 한국연구재단 과제(RS-2025-21882984) 연구 결과의 일부임.

**경상국립대학교 토목공학과 공학석사(주저자: s2125410@naver.com)

***경상국립대학교 토목공학과 공학연구원·교수(교신저자: lskang@gnu.ac.kr)

서 향후 유지관리단계의 핵심 기술이 될 수 있다.

그렇지만 여전히 구조물에 있는 실제 균열의 크기와 형태 등의 손상정보관리는 일차적으로 2D이미지의 사진정보로 관리되는 사례가 많은 실정이다. 2D 이미지 사진에 있는 균열 등의 정보를 3D기반으로 관리하기 위해서는 동일한 구조물의 3D모델에 정합하는 절차가 필요하다. 또한 이러한 2D좌표와 3D좌표 정보의 정합 과정은 3D모델의 크기와 형상에 따라서 정확도에 차이가 생길 수 있다. 즉, 일반적으로 교량의 교각 구조물은 기초부, 기둥부, 코핑(coping)부로 구성되지만, 설계단계에 교각 구조물은 통상 한개의 단일 3D모델로 만들어진다. 교각 구조물에서 기둥부에 균열이 있을 때, 기둥을 나타내는 2D이미지에 있는 균열좌표를 기둥을 포함한 교각 전체가 하나로 구성된 3D모델에 정합시키는 것이 용이하지 않다. 반면에 교각 구조물이 각 부위별로 3개의 모델로 분할되어 있으면, 기둥부위의 3D모델만을 대상으로 좌표를 정합시키므로 정합의 정확도가 개선될 수 있다. 본 연구에서는 2D이미지의 사진에 있는 손상정보와 3D모델의 손상정보 정합의 정확도 개선을 위해 3D모델 분할 방법론의 적정성을 검증한다. 이를 위해 Revit 소프트웨어의 Dynamo를 활용한 정점 데이터(Vertex data) 알고리즘 방법과 BSP(Binary Space Partitioning), K-D Tree(K-Dimensional Tree) 알고리즘 방법을 비교 검증한다. 3D모델의 X, Y, Z축 분할 검증을 위해 객체를 8등분으로 분할 한 결과를 바탕으로 세 가지 분할 방법론을 비교 분석하고 분할 전후 체적변화 값을 분석하여 적정 객체 분할 방법을 선정한다.

설계단계에 제작된 3D모델은 구조물의 외형중심으로 LoD 200수준의 포괄적 모델이 생성되어 설계 단계 외에는 활용성이 저하되고 있다. 즉, 시공단계의 공정단위 모델이나 유지관리단계의 균열정보 표현을 위한 객체 모델로 활용성이 저하되므로 설계단계에 제작된 3D모델을 시공과 유지관리를 포함하는

생애주기 동안 활용하기 위해서는 초기모델을 사용 목적에 맞도록 분할 할 필요가 있다. 이로써 설계단계 3D모델의 생애주기 동안 재활용율을 개선할 수 있다.

2. 연구 동향

3D모델의 객체 분할 방법론에 대한 연구 동향은 다음과 같다.

Kamari and Ham(2021)는 포인트 클라우드(Point cloud) 모델에서 ROI(Region of Interest)를 자동으로 분할하여 체적을 측정하는 방법을 제안하였으며, Revit의 버전에 따라 API(Application Programming Interface) 함수 제공의 차이로 인한 객체 생성 문제점도 파악되었다. 유영무 외(2021)은 철도 터널 3D 캐드 모델을 포인트 클라우드 데이터로 변환하여 학습 DB를 구축하고 학습 DB에 따라 총 20가지 사례에 대한 터널 객체 분할 인식을 수행하였으며, 새로운 형상의 터널을 대상으로 객체 분할을 수행할 때마다 새로운 학습 DB를 구축해야 하는 문제점이 파악되었다.

Charlest et al.(2017)은 PointNet 알고리즘을 통해 3D 포인트 클라우드 데이터를 대상으로 객체 분류 및 분할의 연구를 수행하였으며, 박승화 외(2015)은 실시설계 수준의 구조 및 마감 물량 산출과 공정별 부위 구분을 위해 복합벽체 자동 분할과 정보추출 알고리즘을 제안하였다. Le and Duan(2017)는 기계 CAD 객체에 대한 새로운 분할 알고리즘에 관한 연구를 진행하였다. 3D모델의 방향성을 분석하여 탐색 공간을 줄이고 2D 데이터인 원과 선을 활용하여 3D객체를 분할하는 연구를 진행하였으며, 원과 선의 2D 데이터는 거친 면을 가진 객체와의 정합에 오류가 발생하기 때문에 분할 정확도가 떨어진다는 한계가 존재하였다.

객체의 정점데이터를 이용한 분할과 관련된 연구

동향은 다음과 같다.

황재영 외(2023)은 설계단계 BIM 모델의 시공단계 재활용을 위한 방안으로 교량 BIM 객체의 코핑과 기둥을 구분 없이 작성하여 정점 데이터를 기반으로 분할하는 방법론을 제시하였으며, 다른 방법과의 비교검증을 하지 않았으므로 분할 정확도의 검증이 어렵다. 김대영 외(2009)는 적응형 정점 군집화를 통해 메쉬의 컴팩트성을 유지하기 위해 정점 분포를 고려하여 분할 축을 동적으로 설정해 다수의 파티션으로 객체를 분할하는 연구를 진행하였으며, 기존의 정점 군집화 알고리즘과 질적 비교를 위한 측정 방법이 존재하지 않는 문제점이 있다. 이승수 외(2014)는 도로나 철도와 같은 선형기반 시설 공종의 경우 BIM 소요 정보와 세분화된 연계를 위해 정점 데이터를 생성할 수 있는 서피스모델 분절 객체화 모듈에 관한 연구를 진행하였으며, 모듈에 관한 설계 방법론을 제시하였으나 실제 다양한 객체 분할은 진행하지 않았다는 한계점이 있다. Longjiang et al.(2013)는 MRF(Markov Random Field)를 사용하여 3D 메쉬 정점 분할 알고리즘의 성능을 개선하기 위해 메쉬 정점이 분할 경계에 놓였을 때, 3D 메쉬 데이터의 표면을 다시 분할한 후 제공된 정점 경계의 변화를 분석하는 연구를 진행하였으며, 다양한 데이터 셋에 대해서는 추가적인 검증이 필요하다는 한계점이 존재하였다.

기존의 알고리즘 기반 객체 분할 연구에서는 포인트 클라우드를 이용하여 분할하는 연구가 대부분으로 타 방법과의 정확도 비교가 부족한 부분이 있고, 객체 생성과 DB를 객체마다 구성해야 한다는 한계점이 존재하고 있다. 또한 정점데이터를 이용한 객체 분할 연구에서도 객체를 분할하는 프로그램과 기존의 객체 분할 알고리즘을 비교 검증하는 연구가 부족한 한계가 존재하고 있다. 본 연구에서는 이러한 한계점을 개선하기 위해 정점데이터와 공간좌표 기반의 3개 분할 방법론을 비교 검증하여 각 방법의

분할 정확성을 파악한 후 3D모델의 최적 분할 방법론을 제시한다.

3. 3D모델 객체 분할 방법론

3.1 정점데이터기반 객체 분할

정점데이터기반 객체 분할은 Fig. 1과 같이 좌표 X, Y 값의 최댓값과 최솟값의 차이로 생성된 Bounding Box를 하나의 객체로 구성된 3D모델에 사용자가 원하는 X, Y, Z축 방향으로 생성하고, 생성된 Bounding Box를 기준으로 객체를 분할하여 여러 개의 객체로 재구성하는 것을 의미한다(황재영 외, 2023).

정점데이터기반 객체 분할 방법론은 3D모델의 정점 추출 및 데이터화, 군집화 알고리즘 적용, 분할면 생성, 객체분할 및 매개변수 생성 순으로 구성되며, 각 정점의 위치 데이터를 추출하는 과정이 필요하고 연구에서는 Revit Dynamo 노드를 활용하여 객체의 정점 데이터를 추출한다.

3.2 BSP 알고리즘 객체 분할

BSP 알고리즘은 Shumacker에 의해 제시되었으며 3D그래픽에서 공간을 효율적으로 분할하는 기법이다(Schumacker et al., 1969). 게임에서는 특히 1

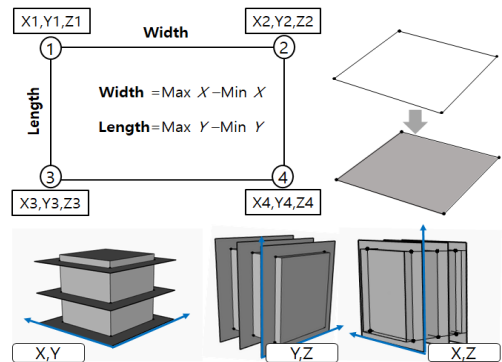


Fig. 1. Vertex Data Based Object Segmentation Method

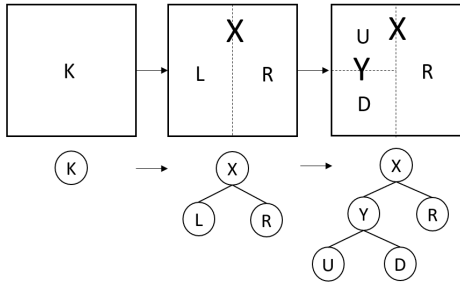


Fig. 2. BSP Algorithm Segmentation Method

Source: Kim and Lee (2011) (in Korean)

인칭 슈팅 게임과 같은 3D 컴퓨터 게임 제작에서 널리 사용된다. BSP 알고리즘은 Fig. 2와 같이 평면상의 블록 집합으로 객체를 분할하는 기법으로 이러한 분할 과정에서 BSP Tree라 불리는 트리구조가 생성된다. 그리고 BSP Tree를 통해 하나의 공간을 특정한 최종 목적을 만족할 때까지 작업 공간을 분할하는 알고리즘이다. BSP Tree는 작업 공간 내에서 볼륨을 복셀로 분할하는 최대 반복 레벨을 정의하고, 하나의 복셀이 가질 수 있는 최대 삼각 폴리곤 (Polygon)의 수를 정의한다(김용관·이강희, 2011). 또한 BSP Tree는 객체를 전체로 하는 경계 볼륨을 사각형으로 하여 가장 긴 부분을 분할 면으로 분할한다(남승우·정연철, 2002).

3.3 K-D Tree 알고리즘 객체 분할

K-D Tree 알고리즘은 Bentley에 의해 제시된 다차원 이진 검색트리 구조이다(Bentley, 1975). K-D

Tree 알고리즘은 다른 알고리즘에 비해 상대적으로 빠른 처리 시간을 가졌고 메쉬를 효율적으로 분할하는데 사용되거나(김대영·이혜영, 2011), 광선과 다각형 간의 교차 테스트를 효과적으로 수행하기 위해 사용되는 기법이다(강윤식 외, 2008). K-D Tree 알고리즘은 Fig. 3과 같이 객체의 전체적인 데이터 분포를 통해 데이터가 절반이 될 때 평면을 삽입하여 분할을 진행한다. 재귀적 방식을 사용하여 노드(Node)의 개수보다 분할된 객체의 개수가 같거나 커질 때까지 반복하여 균등한 크기의 객체로 분할될 수 있게 한다(정재화, 2015).

4. 객체 분할 방법론의 비교 검증절차

4.1 알고리즘별 객체 분할 검증 절차

연구에 적용한 객체 분할 알고리즘들에 대한 실험 및 검증 방법은 Fig. 4와 같다.

객체 분할의 정확성을 비교하기 위해 Fig. 4와 같이 BIM 기반 프로그램인 Revit을 기반으로 제작된 교량의 기둥 객체와 코핑 객체를 분할 대상으로 선정 후, 각 방법론으로 분할하고 분할된 객체의 부피를 원 객체의 부피와 비교하는 방법을 적용하였다.

정점데이터 기반 분할은 기존의 3D모델 객체의 좌표를 Dynamo Script를 제작하여 추출한다. 추출된 정점 좌표를 기준으로 분할 면을 생성한 뒤 분할 면의 방향대로 객체를 분할한다. BSP 알고리즘을 통

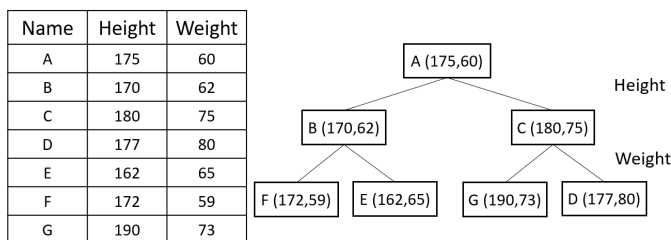


Fig. 3. K-D Tree Algorithm Segmentation Method

Source: Lee (1997) (in Korean)

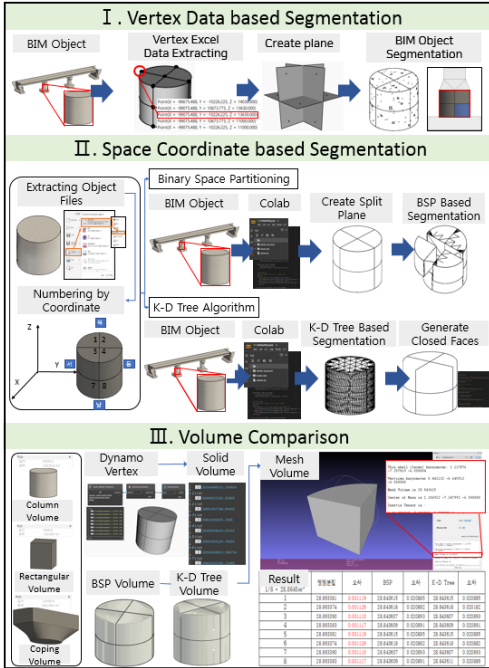


Fig. 4. Comparison Verification Process for 3D Model Segmentation Algorithms

한 분할은 Revit에서 obj 형식으로 분할 대상 객체 파일을 추출하고 추출된 객체를 BSP 알고리즘을 통해 해당 객체에 X, Y, Z축을 기준으로 분할 면을 생성한 후 생성된 분할 면을 기준으로 객체를 8등분으로 분할한다. 객체의 분할 수는 분할 대상 모델의 규모에 따라 정할 수 있으며, 본 사례에서는 교각을 수직 축으로 2등분한 후 수평축으로 4등분하여 8등분으로

구성한다. K-D Tree 알고리즘을 통한 분할은 분할 대상 객체 파일을 추출하고 추출된 객체에 K-D Tree 알고리즘을 적용하여 X, Y, Z축을 기준으로 Z값이 0에 도달할 때까지 반복적으로 분할을 진행한 뒤 객체를 8개로 분할한다.

정점 데이터 기반 객체 분할과 BSP, K-D Tree 알고리즘을 통해 분할된 객체의 정확성을 비교하기 위해 Dynamo Node를 통해 분할된 객체의 부피를 측정한다. 또한 알고리즘을 통해 분할된 객체의 부피 측정을 위해 MeshLab 소프트웨어를 이용해 측정한다. 원 객체의 부피를 8등분으로 나눈 값과 정점 데이터 및 두 가지 알고리즘을 통해 분할된 객체의 부피를 비교하여 정확성을 검증하고 3D모델 객체 분할에 적합한 알고리즘을 선정한다.

4.2 알고리즘별 교각 모델의 분할

4.2.1 정점데이터기반 교각 모델 분할

정점데이터기반 교각모델의 분할은 Fig. 5와 같이 세 가지 분할 대상 3D모델의 정점 좌표를 Dynamo Script의 Vertex Node를 통해서 Excel 형태로 추출한다. 그리고 분할하려는 3D모델을 Geometry Node를 사용하여 Dynamo 환경에 가져온다. 다음으로 분할 대상 교각모델에 대한 Excel 형태로 추출된 좌표 값을 기반으로 분할 구역을 설정하기 위해 Rectangle Node를 사용하여 각각 XY, XZ, YZ 평면을 기준으로

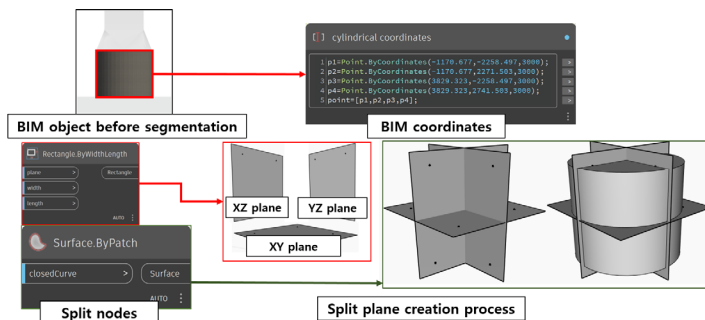


Fig. 5. Object Segmentation Method of Pier Model Using Vertex Data

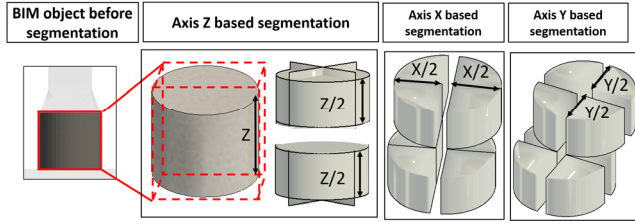


Fig. 6. Object Segmentation Method of Pier Model Using BSP Algorithm

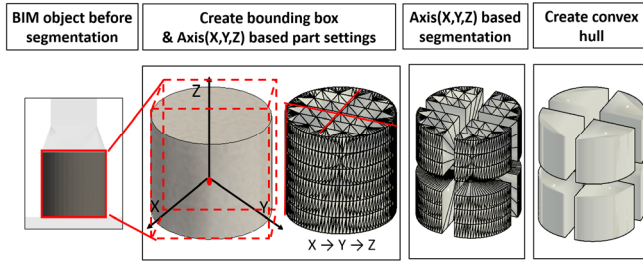


Fig. 7. Object Segmentation Method of Pier Model Using K-D Tree Algorithm

3개의 분할 면을 생성한다. 생성된 면을 기준으로 교각 모델을 분할하기 위해 Split Node를 사용하여 분할 대상 교각모델을 8등분으로 분할한다.

4.2.2 BSP 알고리즘 교각 모델 분할

BSP 알고리즘을 통해 세 가지 객체를 분할하기 위해 Fig. 6과 같이 각 객체의 경계를 계산해 X, Y, Z축의 최대 크기에서 Z축 기준으로 중심 위치에 각 객체의 중심 모양에 알맞은 수평 분할 면을 생성하고 X, Y축을 기준으로 각 객체에 알맞은 수직 분할 면을 생성한 후, 생성된 분할 면을 기준으로 BSP 알고리즘을 적용하여 분할한다. BSP 알고리즘은 객체를 공간적인 트리구조로 나누는 기법으로 객체의 수직 분할면을 기준으로 Upper_Part의 객체 상단 부분과 Lower_Part의 객체 하단부분으로 분할한다. 상단과 하단으로 분할된 객체를 X축 기준으로 수직인 평면과 Y축 기준으로 수직인 평면을 기준으로 객체가 각 4등분으로 분할될 때까지 객체를 분할하여 총 8등분의 객체를 생성한다.

4.2.3 K-D Tree 알고리즘 교각 모델 분할

K-D Tree 알고리즘을 통해 교각 모델을 분할하기 위해 Fig. 7과 같이 각 단계에서 객체의 메쉬 Bounding Box를 생성한 뒤 X, Y, Z축의 중간값을 계산하고, 중간값을 기준으로 각 객체를 두 부분으로 나눈다. 축방향으로 중간값보다 큰 영역은 positive_part로 축방향으로 하고 중간값보다 작은 영역은 negative_part로 설정하여 재귀 호출을 통해 분할을 반복한다. 각 분할된 부분에 대해 객체의 Z축을 기준으로 depth를 감소시켜 종료 조건을 설정한다. depth가 설정된 Z축이 0이 되는 값에 도달하면 더 이상 분할을 진행하지 않고 현재 메쉬 상태를 그대로 변환하여 총 8등분의 객체를 생성한다. 마지막으로 생성된 객체의 부피 측정을 위해 외부구조를 감싸는 Convex Hull을 생성한다.

4.3 분할 교각 모델의 부피 비교

분할된 교각 모델의 분할 정확성을 검증하기 위해 정점데이터, BSP, K-D Tree방법에 의해 원기둥, 사각기둥, 코핑 객체를 각각 8등분으로 분할한 후 Revit

의 Dynamo Node와 MeshLab 소프트웨어를 통해 부피를 측정하였다.

정점데이터기반 분할 방법론을 통해 분할된 객체의 부피를 측정하기 위해 분할된 객체를 Dynamo에 불러온다. 그 후 객체의 속성 정보를 추출하는 Node인 Geometry Node를 통해 객체의 속성 정보를 추출한 후 Solid Volume Node를 통해 객체의 부피를 측정한다. BSP와 K-D Tree 알고리즘을 통해 분할된 객체는 MeshLab을 실행하여 Import Mesh를 통해 모델을 업로드한 후 Quality Measures and Computations 기능의 Compute Geometric Measures 방법을 통해 부피를 측정한다. 이러한 비교 값을 통해 검증된 오차의 크기가 가장 작은 분할 방법론이 3D 모델의 객체 분할에 적합하다고 판단할 수 있다.

5. 교각모델 객체 분할의 정확성 비교

5.1 정점데이터기반 객체 분할 검증

본 연구에서는 교량 기둥 부위의 대표적 종류인 사각형, 원형 기둥 형상과 코핑을 대상으로 객체 분할 정확성의 비교검증을 시도하였다. 먼저 정점데이터를 사용하여 원기둥, 사각기둥과 코핑을 분할 한 결과는 Fig. 8과 같다. 객체들의 중심 좌표를 기준으로 XY, XZ, YZ 축으로 생성된 분할 면에 따라 기둥 객체가 분할된 것을 확인할 수 있다. 또한 분할된 객체들에 대해 각각의 부피를 측정하기 위해 Dynamo를 사용하여 추출한 결과는 Table 1과 같다. 원기둥의 경우 가장 작은 값이 28.863374m³로 측정됐고 가장 큰 값은 28.863390m³로 측정되었다. 사각기둥의 가장 작은 값은 18.749994m³, 가장 큰 값은 18.750005m³로 측정되었다. 마지막으로 코핑 상부 객체의 경우 가장 작은 값이 39.062494m³, 가장 큰 값은 39.062505m³로 측정되었다. 코핑 하부객체의 경우 26.498253m³이 가장 작은 값으로, 26.564400m³이 가장 큰 값으로 측정되었다.

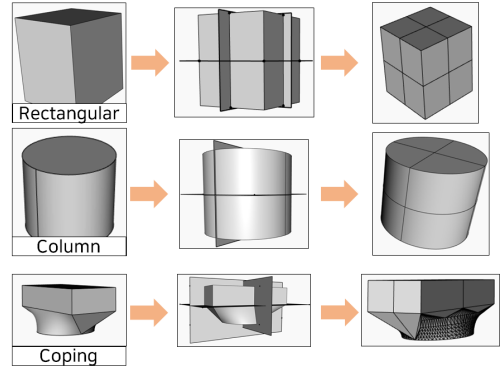


Fig. 8. Vertex Data Based Object Segmentation Result

Table 1. Vertex Data Based Object Segmentation Result

Column (m ³)	Rectangle (m ³)	Coping (m ³)
28.863381	18.749998	39.062497
28.863374	18.749994	39.062505
28.863390	18.750005	39.062494
28.863383	18.750005	39.062502
28.863381	18.749998	26.564400
28.863374	18.749994	26.498253
28.863390	18.750005	26.498358
28.863383	18.750001	26.564369

5.2 공간좌표 알고리즘 객체 분할 검증

BSP 알고리즘과 K-D Tree 알고리즘을 통해 3D형태 교각모델의 원형 기둥과 사각형 기둥, 코핑 객체를 8등분으로 분할한 결과는 Fig. 9와 Fig. 10과 같다.

분할된 각 객체의 부피를 측정하기 위해 MeshLab 프로그램에 업로드하여 Quality Measures and Computations 기능을 통해 부피를 측정한 결과는 Table 2, Table 3과 같다. BSP 알고리즘의 경우 원기둥의 경우 가장 작은 값이 28.843607m³로 측정되었고 가장 큰 값으로는 28.843618m³로 측정되었다. 사각기둥에서는 모든 객체가 18.750000m³로 균등하게 측정된 것을 알 수 있었다. 코핑의 경우 상부 객체에서는 가장 작은 값이 39.062500m³, 가장 큰 값

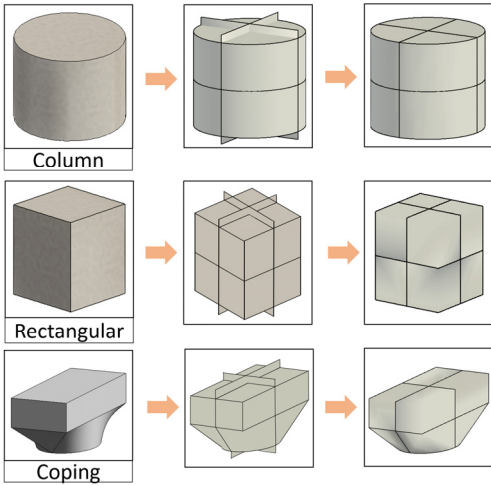


Fig. 9. BSP Algorithm Object Segmentation Result

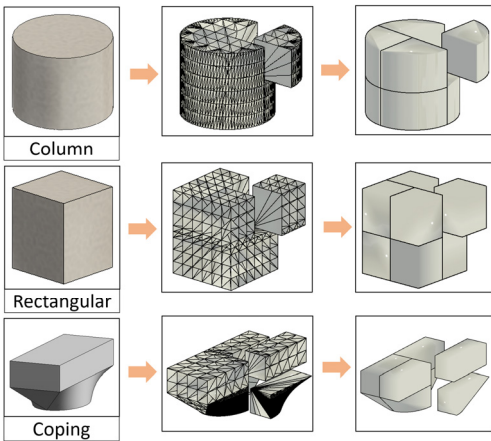


Fig. 10. K-D Tree Algorithm Object Segmentation Result

은 39.062504m³로 측정되었다. 또한 하부 객체에서는 가장 작은 값이 28.901646m³, 가장 큰 값은 28.901648m³로 측정되었다.

K-D Tree 알고리즘의 경우 원기둥의 경우 가장 작은 값으로 28.843607m³, 가장 큰 값으로는 28.843618m³이 측정되었다. 사각기둥에서는 가장 작은 값으로 18.749998m³, 가장 큰 값으로는 18.750006m³이 측정되었다. 코핑의 경우 상부 객체에서는 가장 작은 값

Table 2. BSP Algorithm Based Object Segmentation Result

Column (m ³)	Rectangle (m ³)	Coping (m ³)
28.843615	18.750000	39.062504
28.843618	18.750000	39.062500
28.843607	18.750000	39.062504
28.843609	18.750000	39.062500
28.843615	18.750000	28.901646
28.843618	18.750000	28.901646
28.843607	18.750000	28.901648
28.843609	18.750000	28.901646

Table 3. K-D Tree Algorithm Based Object Segmentation Result

Column (m ³)	Rectangle (m ³)	Coping (m ³)
28.843615	18.750002	39.062500
28.843618	18.750000	39.062504
28.843607	18.750002	39.062500
28.843609	18.749998	39.062500
28.843615	18.750002	28.901644
28.843618	18.750000	28.901648
28.843607	18.750002	28.901644
28.843611	18.749998	28.901646

으로 39.062500m³, 가장 큰 값으로는 39.062504m³이 측정되었다. 하부 객체에서는 가장 작은 값으로 28.901644m³, 가장 큰 값으로는 28.901648m³이 측정되었다.

5.3 객체 분할 부피 비교 검증 결과

Table 4, Table 5, Table 6과 같이 3D모델의 분할 전 원기둥, 사각기둥, 코핑의 전체 부피 크기와 8등분 분할 후 부피 값을 각 방법론을 통해 비교하고 오차를 추출하였다. 원기둥에서는 정점데이터 기반 분할 방법이 평균 0.001118m³으로 가장 작은 오차를 나타냈다. 사각기둥에서는 BSP 알고리즘을 이용한 분할 방법이 오차가 0으로 정확하게 분할된 것을 알

Table 4. Column Segmentation Experiment Result Error

Column (230.916 m ³) 1/8=28.8645 m ³	Vertex	Error	BSP	Error	K-D Tree	Error
1	28.863381	0.001119	28.843615	0.020885	28.843615	0.020885
2	28.863374	0.001126	28.843618	0.020882	28.843618	0.020182
3	28.863390	0.001110	28.843607	0.020893	28.843607	0.020893
4	28.863383	0.001117	28.843609	0.020891	28.843609	0.020891
5	28.863381	0.001119	28.843615	0.020885	28.843615	0.020885
6	28.863374	0.001126	28.843618	0.020882	28.843618	0.020882
7	28.863390	0.001110	28.843607	0.020893	28.843607	0.020893
8	28.863383	0.001117	28.843609	0.020891	28.843611	0.020889

Table 5. Rectangle Segmentation Experiment Result Error

Rectangle (150 m ³) 1/8=18.75 m ³	Vertex	Result	BSP	Result	K-D Tree	Result
1	18.749998	0.000002	18.750000	0	18.750004	0.000004
2	18.749994	0.000006	18.750000	0	18.750004	0.000004
3	18.750005	0.000005	18.750000	0	18.750006	0.000006
4	18.750005	0.000005	18.750000	0	18.749998	0.000002
5	18.749998	0.000002	18.750000	0	18.750004	0.000004
6	18.749994	0.000006	18.750000	0	18.750004	0.000004
7	18.750005	0.000005	18.750000	0	18.750006	0.000006
8	18.750001	0.000001	18.750000	0	18.749998	0.000002

Table 6. Coping Segmentation Experiment Result Error

Coping (262.404 m ³) Top (156.25 m ³) Bottom (106.154 m ³)	Vertex	Result	BSP	Result	K-D Tree	Result
1	39.062497	0.000003	39.062504	0.000004	39.062500	0.000006
2	39.062505	0.000005	39.062500	0	39.062504	0.000004
3	39.062494	0.000006	39.062504	0.000004	39.062500	0.000005
4	39.062502	0.000002	39.062500	0	39.062500	0.000004
5	26.564400	0.025900	28.901646	2.363146	28.901644	2.363144
6	26.498253	0.040247	28.901646	2.363146	28.901648	2.363148
7	26.498358	0.040142	28.901648	2.363148	28.901644	2.363144
8	26.564369	0.025869	28.901646	2.363146	28.901646	2.363146

수 있다. 또한 사각으로 생성된 코핑 상부에서는 BSP 알고리즘을 이용한 분할 방법이 평균 0.00004m³, 곡선형 사다리꼴로 생성된 하부객체에서는 정점 분

할 방법이 평균 0.033039m³으로 가장 작은 오차를 나타냈다.

사각형 객체에서 BSP 알고리즘에 오차가 가장 작

은 원인은 BSP 알고리즘은 생성된 평면을 기준으로 직선과 평면을 이용해서 객체를 분할하기 때문에 곡선이 존재하지 않는 사각형 객체에서 비교적 정확한 것을 알 수 있다. 하지만 곡선이 존재하는 원기둥과 코핑의 하부객체에서는 오차가 큰 것을 파악할 수 있다. 정점데이터 기반 분할은 원기둥과 코핑 하부 객체에서 가장 높은 정확성을 나타냈고 사각기둥에서는 오차가 생성되지 않은 BSP 알고리즘 다음으로 정확한 것을 알 수 있다.

6. 결론

BIM모델 적용이 보편화되면서 설계단계부터 3D 모델이 제작되는 사례들이 증대되고 있으나 시공단계와 유지관리단계에 사용되는 3D모델의 LoD(Level of Detail)차이로 인하여 생애주기 활용성이 저하되고 있다. 이러한 생애주기 활용성을 높이기 위해서는 3D모델을 사용 목적에 따라 부위 또는 공정단위로 분할하는 과정들이 요구된다. 본 연구에서는 3D 모델 분할 방법의 정확도를 검증하기 위해 보편적으로 많이 사용하는 정점데이터기반 분할과 BSP 및 K-D Tree 알고리즘기반 분할 방법을 적용하여 모델의 사용 목적에 맞는 적절한 분할 방법을 제시하였다. 이를 위해 설계단계에서 생성된 교각 모델 객체의 원기둥, 사각기둥, 코핑을 대상으로 각 방법론을 적용하여 각 객체를 8등분으로 분할한 후 분할 전후의 부피에 대한 오차를 측정하여 정확성을 비교하였다.

원기둥에서는 평균 $0.001118m^3$ 으로 정점 데이터기반 분할 방법이 가장 작은 오차를 나타냈다. 사각기둥에서는 평균 0으로 BSP 알고리즘을 이용한 분할 방법이 가장 작은 오차를 나타내었으며, 코핑의 하부객체에서는 $0.033039m^3$ 으로 정점 데이터기반 분할 방법이 가장 작은 오차를 나타냈다. 직선과 면을 기반으로 분할하는 BSP 알고리즘이 사각형 객체인 사각기둥과 코핑 상부에서 높은 정확성을 보

였다. 하지만 곡선 형태인 원기둥과 코핑 하부에서는 정점 데이터기반 분할 방법에 비해 높은 오차를 나타냈다. 정점 데이터기반 분할 방법은 원형 객체에서 가장 높은 정확성을 나타내었고, 사각형 객체에서도 K-D Tree보다 높은 정확성을 나타냈다.

전체 분석 결과에서 사각형모델에서는 BSP알고리즘이 적정하고 원형모델과 코핑형태 모델에서는 정점데이터기반의 분할 방법이 적정한 것으로 파악되었다. 이러한 결과에서 사각형 형상 모델의 분할 시에는 BSP알고리즘 적용이 적절하고, 원형과 일부 곡선을 포함한 형상 모델의 분할에는 정점데이터기반 분할 방식이 적정함을 확인할 수 있다.

연구에서는 교각 모델에서 원기둥과 사각기둥, 코핑 3개의 객체를 대상으로 정확성을 비교하였기 때문에 원형과 사각형이 아닌 비정형 형상을 갖는 객체에 대해서는 추가적인 사례 분석이 필요하고, 체적비교 외의 형상적 외관 비교 알고리즘 구성과 체적 비교를 위해 객체의 분할 개수를 증감시켜 검증하면 정확도의 편차를 파악할 수 있다.

참고문헌

1. 강윤식·박우찬·서충원·양성봉(2008), "GPU상에서 동작하는 Ray Tracing을 위한 효과적인 k-D tree 탐색 알고리즘", 『정보과학회논문지: 시스템 및 이론』, 35(3): 133~140.
2. 김대영·김종원·이혜영(2009), "적응형 정점 군집화를 이용한 메쉬 분할", 『한국컴퓨터그래픽스학회논문지』, 15(3): 19~26.
3. 김대영·이혜영(2011), "외부 메모리에 있는 대용량 메쉬 분할 알고리즘", 『정보과학회논문지: 시스템 및 이론』, 38(4): 186~196.
4. 김용관·이강희(2011), "BSP 기능을 활용한 3차원 가속 렌더링에 관한 연구", 『애니메이션연구』, 7(2): 29~40.
Kim, Y. G. and K. Lee (2011), "Study on Accelerated 3D Rendering Using BSP Functions", *The Korean Society of Animation Studies*, 7(2): 29~40 (in Korean).

5. 남승우·정연철(2002), “공간분할트리를 이용한 효율적인 충돌탐지 방법에 관한 연구”, 「한국정보처리학회 추계학술발표대회 논문집」, 9(2): 11~14.
6. 박승화·김홍수·윤두영(2015), “실시설계단계에서 수량산출을 위한 복합벽체 자동분할에 관한 연구”, 「한국CAD/CAM학회논문집」, 20(2): 124~132.
7. 양다현·이민진·안효준·정현진·이종한(2023), “교량 바닥판의 균열 BIM 생성 및 BIM-FEM 상호 연계 알고리즘 개발”, 「대한토목학회논문집」, 43(6): 689~698.
8. 유영무·김병규·박정준(2021), “딥러닝을 활용한 철도 터널 객체 분할에 학습 데이터가 미치는 영향”, 「한국지반공학학회논문집」, 37(11): 107~118.
9. 이구남(1997), “화일조직을 위한 인덱싱 기법의 성능 특성 비교”, 「정보교육학회논문지」, 1(1): 49~59.
Lee, G. N. (1997), “Some Characteristics of the Performance in Comparison with Indexing Techniques for File Organization”, *Association of Information Education*, 1(1): 49~59 (in Korean).
10. 이상호·이정빈·탁호균·이상호(2023), “BIM-COBie를 활용한 교량상부구조의 손상정보 관리 방법”, 「대한토목학회논문집」, 43(2): 165~173.
11. 이상수·이민주·정중윤·서종원(2014), “선형기반 시설물을 위한 Flexible BIM 시스템의 설계”, 「대한토목학회논문집」, 34(2): 677~685.
12. 정재화(2015), “데이터 샘플링을 통한 각 기반 공간 분할 병렬 스카이라인 질의처리 기법”, 「컴퓨터교육학회 논문지」, 18(5): 63~70.
13. 최용규·Pa Pa Win Aung·산육타 아라비카·차기춘·박승희(2024), “BIM 모델과 증강현실을 활용한 교량 유지관리방안 연구”, 「대한토목학회논문집」, 44(1): 1~9.
14. 황재영·이재희·강인석(2023), “정점데이터기반 분할기법을 활용한 BIM모델의 활용 방안 연구”, 「LHI Journal」, 14(1): 123~134.
15. Bentley, J. J. (1975), “Multidimensional Binary Search Trees Used for Associative Searching”, *Communications of the ACM*, 18(9): 509~517.
16. Qi, C. R., H. Su, M. Kaichun and L. J. Guibas (2017), “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation”, *Proceedings in 2017 IEEE conference on computer vision and pattern recognition (CVPR)*, 652~660.
17. Kamari, M. and Y. Ham (2021), “Vision-Based Volumetric Measurements via Deep Learning-Based Point Cloud Segmentation for Material Management in Jobsites”, *Automation in Construction*, 121: 1~13.
18. Le, T. and Y. Duan (2017), “A Primitive-Based 3D Segmentation Algorithm for Mechanical CAD Models”, *Computer Aided Geometric Design*, 52~53: 231~246.
19. Longjiang, E., S. Waseem and A. Willis (2013), “Using a MAP-MRF Model to Improve 3D Mesh Segmentation Algorithms”, *2013 Proceedings of IEEE Southeastcon*, 2013: 1~7.
20. Schumacker, R. A., B. Brand, M. G. Gilliland and W. H. Sharp (1969), “Study For Applying Computer-Generated Images to Visual Simulation”, *Air Force Human Resources Laboratory, Air Force Systems Command*, 1~151.

요 약

설계단계에 3D모델이 제작되는 사례가 증대되고 있으나 시공과 유지관리단계에 사용되는 3D모델과의 LoD차이로 인하여 생애주기 활용성이 저하되고 있다. 설계단계에 제작된 외형위주의 3D모델을 시공단계 공정관리와 유지관리단계 손상정보관리에 사용하기 위해서는 3D모델의 분할절차가 필요하다. 본 연구에서는 3D모델의 분할을 위해 정점데이터 활용방법과 BSP 및 K-D Tree 알고리즘 방법을 비교하여 분할의 정확도를 검증하여 적정 분할방법론을 제시한다. 이를 위해 교량의 교각모델을 대상으로 원기둥, 사각기둥, 코핑 객체를 추출하여 각 방법론을 통해 적정 등분으로 분할한 후 분할전후의 부피 비교를 통해 분할방법론의 정확도를 비교하였으며, 정점 데이터 기반의 분할 방법이 원기둥과 코핑 하부에서 가장 작은 오차를 나타내었고 사각기둥과 코핑 상부에서 차순위로 작은 오차를 나타내어 3D 모델의 분할 방법론으로 적정함을 확인하였다.

주제어: 정점데이터, BSP, K-D Tree, 3D모델분할