

# 웹기반 GIS 플랫폼 상 가시화 처리를 위한 대용량 BIM 데이터의 경량화 알고리즘 제시

## A Study on Light-weight Algorithm of Large scale BIM data for Visualization on Web based GIS Platform

김지은\* · 홍창희\*\*

Ji Eun Kim · Chang Hee Hong

**요 약** BIM 기술은 기존 2D 기반 도면처리에서 나아가 3D 모델링을 통한 시설물의 전 생애주기에 발생하는 데이터를 포함한다. 이러한 특성상 하나의 건물은 그 데이터의 방대한 양으로 엄청난 크기의 파일을 생산한다. 대표 표준포맷인 IFC가 그 예로, 객체 기반의 형상정보 및 속성정보를 기반으로 상당한 데이터를 포함한 대용량 처리에 대한 이슈가 종종 발생하고 있다. 이는 렌더링 속도를 증가시키고, 그래픽 카드 용량을 많이 차지하기 때문에, 화면 가시화 측면에서 비효율적이다. 대용량 데이터의 경량화 문제는 프로그램의 프로세스와 품질 측면에서 필수적으로 해결되어야 한다. 본 연구는 국내 및 해외 연구사례에서 경량화에 관련된 다양한 시도를 확인하였다. 이를 기반으로 대용량 BIM 데이터를 효과적으로 컨트롤하고 가시화하기 위해, BIM 특성을 고려하여 최대한 활용할 수 있는 데이터의 경량화 기법을 제안하고 검증하였다. 이는 웹 기반 GIS 플랫폼 상에서 대용량 시설물 데이터를 운용하는데, 최적의 시설물 유형을 분석하고 객체 기반의 IFC 특성을 최대한 활용하여 사용자 측면에서 화면전환의 품질을 확보하고 프로세스 측면에서 효과적인 메모리 운영을 확인하였다.

**키워드** : 건물정보모델링(BIM), 데이터 경량화, 지리정보시스템(GIS), IFC, 가시화

**Abstract** BIM Technology contains data from the life cycle of facility through 3D modeling. For these, one building products the huge file because of massive data. One of them is IFC which is the standard format, and there are issues that large scale data processing based on geometry and property information of object. It increases the rendering speed and constitutes the graphic card, so large scale data is inefficient for screen visualization to user. The light weighting of large scale BIM data has to solve for process and quality of program essentially. This paper has been searched and confirmed about light weight techniques from domestic and abroad researches. To control and visualize the large scale BIM data effectively, we proposed and verified the technique which is able to optimize the BIM character. For operating the large scale data of facility on web based GIS platform, the quality of screen switch from user phase and the effective memory operation were secured.

**Keywords** : Building Information Modeling, Data light weighting, Geographic Information System, Industry Foundation Classes, Visualization

## 1. 서 론

### 1.1 연구배경 및 목적

최근 차세대 건설공간정보 기술의 필요성에 대한 인식이 증가함에 따라, 건설 분야의 BIM 데이터를 활용한 3차원 공간정보 구축을 위한 의미 있는 연구가

국내외적으로 이루어지고 있다. 그러나 BIM (Building Information Modeling)과 GIS (Geographic Information System) 간의 서로 다른 데이터 스키마로 인해 호환이 어렵고 BIM과 GIS의 상호운용을 위한 표준이 마련되어 있지 않은 상황이다. 특히 BIM 기술은 기존 2차원 기반 도면처리에서 나아가 3차원 모델링을 통한 시설물의 전 생애주기에 발생하는 데이터를 포함한다. 이

† This work was researched by a grant from a Strategic Research Project (Development of BIM/GIS Interoperability Open-Platform 2015) funded by the Korea Institute of Construction Technology.

\* Ji Eun Kim, Researcher, ICT Convergence and Integration Research Institute, Korea Institute of Civil engineering and building Technology, jekim@kict.re.kr (Corresponding Author)

\*\* Chang Hee Hong, Senior Researcher, ICT Convergence and Integration Research Institute, Korea Institute of Civil engineering and building Technology, chhong@kict.re.kr

러한 특성상 하나의 건물은 그 데이터의 방대한 양으로 엄청난 크기의 파일을 생산한다. 대표 표준포맷인 IFC (Industry Foundation Classes)가 그 예로, 객체 기반의 형상정보 및 속성정보를 기반으로 상당한 데이터를 포함한 대용량 처리에 대한 이슈가 종종 발생하고 있다. 이는 렌더링 속도를 증가시키고, 그래픽 카드 용량을 많이 차지하기 때문에, 화면 가시화 측면에서 비효율적이다.

대용량 데이터의 경량화 문제는 프로그램의 프로세스와 품질 측면에서 필수적으로 해결되어야 한다. 이와 관련하여 국내외 연구사례에서 다양한 대안을 찾고자 하는 시도를 확인하였다. 본 연구는 대용량 BIM 데이터를 효과적으로 컨트롤하고 가시화하기 위해, BIM 특성을 고려하여 최대한 활용할 수 있는 데이터의 경량화 기법을 제안하고 검증하였다.

### 1.2 연구흐름도

본 연구의 프로세스는 다음과 같다. 2장에서는 대용량 3차원 데이터의 경량화 기법과 관련하여 국내외 연구 사례를 조사하고 분석하였다. 3장에서는 본 연구의 기반이 되는 BIM/GIS 상호운용 플랫폼을 살펴보고 형상정보의 데이터 변환 시 발생하는 대용량 처리에 대한 문제점을 파악한 뒤 본 연구에 적합한 기법을 찾아 소개하였다. 4장에서는 상기 내용을 바탕으로 대용량 BIM 데이터의 경량화를 위한 알고리즘을 디자인하고 구조를 설계하였다. 5장에서는 앞서 진행된 내용을 검증하고자 다양한 유형의 IFC 샘플 파일을 대상으로 알고리즘을 테스트하였고 결과를 분석하여 마무리 결론을 정리하였다. 연구 프로세스는 Figure 1과 같다.

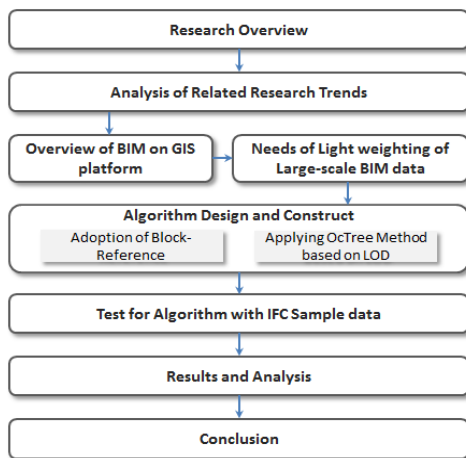


Figure 1. Research Process

## 2. 선행연구 고찰

대부분 3차원 애니메이션을 다루는 게임분야나 의 료용 자료, 문화유산 재건을 위한 레이저 스캐닝 등의 분야에서 주로 대용량 데이터 처리에 대한 연구가 이루어 졌다. 이는 컴퓨터의 메인 메모리조차 컨트롤하기 힘든 것이 사실이다. 이러한 한계를 극복하기 위해 메쉬의 간소화(압축, 분할 등), 외부 메모리 참조 등 다양한 연구가 진행되고 있다.

Park and Lee[9]는 모바일 시스템의 특성을 활용한 원격 렌더링과 지역 렌더링을 함께 진행하여, 대용량 3차원 메쉬에는 각 파티션에 포함되는 점의 수가 균등하도록 분할하는 Kd Tree를 적용하였다. Han[4]은 3차원 포인트 클라우드로부터 OcTree를 생성하기 위하여 DB를 사용하거나 포인트 클라우드 데이터를 메인 메모리에 저장하지 않고 하드디스크 파일에 저장하여 직접적으로 참조하므로써 메모리 참조 방식과 파일 참조 방식을 비교하여 분석하였다. Hongchao Fan et al.[1]은 개구부가 없는 건물의 외피로 이루어진 LOD3 모델로부터 LOD2 모델을 중심으로, CityGML의 3차원 건물 모델을 3단계로 단순화하여 경량화를 진행하였다. Tassilo Glander et al.[3]은 지오메트리 정보를 갖는 식별 가능한 건물 오브젝트, 파사드 텍스처를 갖는 인프라스트럭처 요소, 폴리곤으로 이루어진 비 건물 영역으로 구성된 3차원 블록 셀 기법을 활용하여 유닛에 따라 형상을 일반화하였다. Andre Forberg[2]는 scale-space 이론에 근거하여 모든 직각 건물 구조를 하나의 프로세스에 따라 단순화하고 LOD에 따라 건물 형상이 합쳐질 때까지 평행하게 움직여 3차원 건물 데이터를 단순화하였다. Bo Mao et al.[7]는 LOD 레벨에 따라 3차원 도시모델의 효과적 가시화를 위한 방법으로 다중 형상표현 데이터 구조(multiple representation data structure) 생성을 위한 프레임워크를 개발하고, 프레임워크 내 기능적으로 일부 일반화 방법을 적용하였으며, 이를 가시화하기 위한 효과적인 방법을 개발하였다. Helmut Mayer[8]는 이미지 분석 및 프로세싱을 통한 scale-space 이론에 기반을 두어 2차원과 3차원 상 건물 외곽선의 형상 일반화 베이스 기법 (geometric generalization basing)을 제안하였다.

대용량 데이터의 크기를 줄이는 방법은 상기 조사한 연구들과 같이 다양한 접근이 있다. 기존 Quad-tree, OcTree, R-tree 등의 공간인덱싱 기법들에서 확장하거나 외부 참조 모델을 활용하여 각 특성에 맞는 기준에 따라 메쉬의 간소화를 진행하여 메모리를 축소시키는 방법이 대부분이었다. 이는 모바일, 데스크톱과 같이

작업환경의 특성을 고려하여 일반적인 3차원 노드 데이터 혹은 메쉬 데이터를 가시화하는데 중점을 둔 사례이다. 소수의 BIM 데이터 경량화에 관한 연구 또한 시스템의 특성을 바탕으로 관련 기술을 제시하였다.

BIM 데이터는 건물의 전 생애주기에서 발생하는 다양한 데이터를 모두 포함하기 때문에 가시화를 위한 데이터 규모 축소는 반드시 선행되어야 한다. 본 연구는 플랫폼의 시스템적인 환경보다 메인 데이터포맷인 IFC 데이터에 초점을 두었다. BIM 기반 IFC 데이터의 모델링 기법에서 포인트를 선정하고 이를 확장한 경량화 기법을 제시하였다.

### 3. 데이터 경량화 알고리즘 개요

3장에서는 현재 진행 중인 BIM/GIS 상호운용 개방형 플랫폼 연구를 소개하고 해당 연구의 주요 이슈를 분석하여 데이터 경량화 알고리즘의 등장배경을 설명하였다.

#### 3.1 웹기반 GIS 플랫폼 소개

BIM 기반 건설정보 통합관리 시스템 구축 등과 같은 첨단 건설정보화 기술에 대한 필요성이 인식되면서, 미래 소프트 건설기술 확보 및 세계시장 대응을 위해 국내외 초기 수준의 BIM 정보와 GIS 정보간 상호운용성 확보를 전제로 한 건설공간정보 통합운영 기술의 확보가 요구된다.

본 연구는 한국건설기술연구원에서 진행 중인 “BIM/GIS 기반 건설공간정보 융합기술 개발” 과제의 일환으로, 단일 건물의 실내공간을 다루는 BIM 기술과 도시/지역 단위의 공간을 다루는 GIS 기술을 연계하여 상호운용 가능한 개방형 플랫폼 개발을 진행 중에 있다[5].

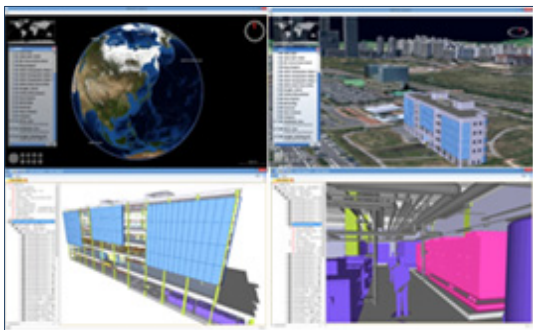


Figure 2. BIM on GIS Platform with IFC Sample data

이는 건설과정에서 발생하는 건설정보(BIM)를 공간정보(GIS)와 융합하여 개별 시설물에서 도시단위까지 복잡하고 지능화된 스마트 시티 실현을 지원하고자 한다. NASA에서 제공한 오픈소스인 WorldWindJava를 활용하여 플랫폼 요구 스펙과 설계를 기반으로, 웹상에서 운영되는 BIM/GIS 상호운용 개방형 플랫폼을 구현하고 관련 요소기술을 개발 중에 있다.

#### 3.2 BIM/GIS 플랫폼 주요 이슈

GIS 플랫폼 상의 상당량의 시설물 데이터를 효율적으로 업로드하기 위하여 본 연구는 전 생애주기에서 발생하는 정보를 포함하는 IFC의 형상정보 데이터와 속성정보 데이터를 분리하여 운영하고자 하였다. 이를 위해 IFC의 형상정보를 GIS 데이터와의 호환을 위해 공간데이터 연계모델의 개념으로 플랫폼 자체 내부 포맷(G3D)을 개발하여 데이터베이스를 구축하였고, 그 과정에서 상기 IFC 파일에서 G3D로 변환하는 과정에서 용량이 증가하는 것을 확인하였다. Figure 3은 IFC 파일을 플랫폼 상에 업로드 하는 과정을 나타낸 개념도이다.

IFC의 경우 3차원 시설물 모델링 시 형상정보가 파라메트릭 데이터 혹은 b-rep으로 저장되지만, G3D 포맷은 IFC의 형상정보를 메쉬 형태로 변환/저장하여 용량이 방대해진다. 이로써 상이한 포맷 간 호환성은 높였으나 용량의 변화로 시설물의 렌더링 프로세스에 걸리는 시간이 길어지고 속도가 증가하는 경향을 확인하였다. 현재는 단일 시설물 업로드에 대한 결과이지만, 향후 BIM/GIS 플랫폼의 특성을 고려했을 때 웹기반에 다양한 크기의 시설물들이 복수로 올라가면 이를 처리할 수 있는 대책을 강구해야 한다.

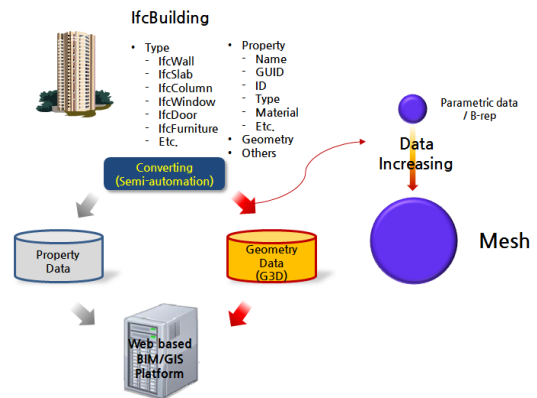


Figure 3. Process of IFC file Uploading

## 4. 데이터 경량화 알고리즘 제시

본 장에서는 앞선 선행연구 고찰을 바탕으로 BIM/GIS 플랫폼 상 대용량 BIM 데이터의 경량화를 위한 알고리즘을 디자인하고 구조를 설계하였다.

### 4.1 경량화 알고리즘 디자인

#### 4.1.1 Block-Reference 기법 도입

대용량 3차원 데이터의 경량화 기법은 앞서 조사한 관련 연구들과 같이 다양한 방법이 있다. 거대한 3차원 메쉬 데이터의 생성이 과거에 비해 비교적 쉽게 이루어지나, 일반 컴퓨터에서도 메인 메모리의 한계로 데이터 전체를 탑재하여 처리하는 것은 아직도 어려운 일이다. 대용량 처리는 경량화뿐만 아니라 이를 통한 효과적인 가시화로 사용자에게 이어져야 한다. 앞서 분석한 LOD 기반의 단순화 알고리즘 혹은 외부 참조 기법 등은 3차원 데이터 가시화를 구체적으로 집중하여 요구하는 분야에 적합하다.

Block-Reference 개념은 Figure 4와 같이 기존 오토캐드에서 많이 활용되고 있는 기능 가운데 하나로, 자주 사용되는 객체 혹은 객체 그룹을 Block으로 정의하여 효율적으로 도면작업에 활용하는 방법이다. BIM 모델링은 일반적으로 건물 혹은 시설물에 종속된 객체를 사전에 라이브러리 형태로 생성 및 저장하여 필요 시 저장된 라이브러리를 호출해 편리하게 사용할 수 있다. 본 연구는 객체 기반의 BIM 모델링 성격을 최대한 활용하고자 하였다. 우선 인공적으로 설계되는 건물 데이터의 특성상, 상당수의 객체가 각각의 고유한 형상정보를 갖지 않는다는 점에 착안하였다. 이는 단순 경량화 기법보다 기반기술의 장점을 활용하여 데이터 원본은 그대로 유지한 채 경량화 프로세스가 가능하다는 장점이 있다. 따라서 본 연구는 경량화의 주된 요소로 Block-Reference를 선정하고, 연구의 특성에 맞춰 알고리즘을 개발하였다.

Block-Reference 기법은 용량을 많이 차지하는 형상정보를 생성할 때 동일한 모양의 형상데이터는 타입이 되는 처음 한 개만 생성하고, 이후 이를 기반으로 각각의 객체가 어떤 형상정보를 어떻게 사용하는지에 대한 정보를 생성하여 참조하는 것이다. 즉 BIM 데이터 전부를 가져와 G3D 포맷으로 변환 후, 복사본에 대한 정보는 참조 DB에서 불러와 가시화 하는 것이다. 이때 생성된 형상정보를 Block이라 정의하고, Block을 참조하고 있는 모든 객체들을 Reference라 정의한다. Figure 5는 BIM 모델링 내 객체 정보가 G3D

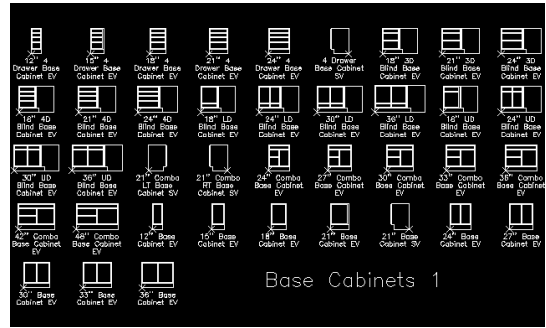


Figure 4. Example drawing for Block of Base Cabinets

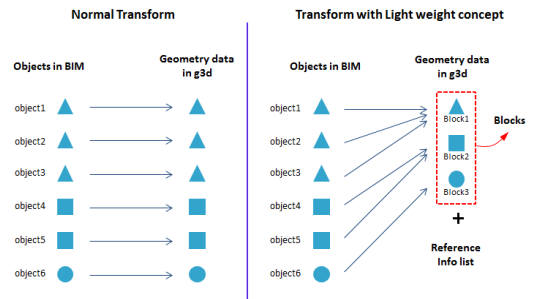


Figure 5. Comparison with 'Normal Transform' and 'Transform with Light Weight Concept'

로 변환되어 저장될 때 일반적인 변환방법과 경량화 기법을 적용하여 변환하는 방법을 비교한 그림이다.

건물 하나의 데이터 패키지는 개략적으로 건물에 대한 정보를 담는 헤더(Header), 동일 형상정보 기반의 대표 타입인 Block, Block을 참조하는 모든 객체들에 대한 Reference 정보로 구성된다. 경량화 기법이 적용되기 이전 G3D 포맷의 경우, IFC로부터 넘어온 모든 형상데이터를 1차적으로 구현했기 때문에 속도나 규모 측면에서 매우 비효율적이었다. 이에 반해 경량화 기법을 적용한 포맷은 상기 기법을 적용시켜 시설물 모델링이 갖는 특성을 고려하여 상당부분 용량을 감소시킬 수 있다.

#### 4.1.2 OcTree 기반 LOD 적용

경량화 알고리즘을 위한 두 번째 개념은 Block-Reference의 데이터 가시화를 위해 공간인덱싱의 한 기법인 LOD 기반 OcTree 를 적용하는 것이다. 아래의 Figure 6은 OcTree를 활용한 공간분할방법을 나타낸 그림이다. 공간인덱싱은 3차원 데이터를 효과적이고 자연스럽게 가시화하는데 하나의 중요한 요소로서, OcTree는 3차원 모델링을 표현하는데 적합하다. Jieun Kim[6]은 대용량 BIM 데이터의 효율적 처리를 위해

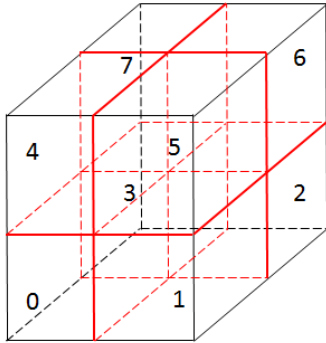


Figure 6. Spatial Indexing Method with OcTree

OcTree 기반의 공간인덱싱 기법을 제안하였다. IFC 스키마 기반의 알고리즘을 구성하고 BIM on GIS 플랫폼 상의 IFC 샘플 데이터에 이를 적용하여 사용자 시점의 경로에 따른 프레임 속도를 측정하였다. 그 결과 알고리즘을 적용한 경우가 초당 3 프레임에서 최대 14 프레임까지 프레임 수가 더 많았으며 동일 시간 대비 보다 많은 데이터를 가시화하며 사용자에게 보다 효과적인 화면을 제공하였다.

본 연구는 앞선 연구의 연장선으로 G3D에 건물의 외부와 내부를 모두 포함하는 LOD를 적용하였다. 외부 LOD에 해당하는 객체들을 모아둔 스킨 Reference 리스트와 건물 내부에서 각 노드에 속한 객체들을 모아둔 노드 Reference 리스트를 생성하여 LOD별 객체를 그룹화하였다.

그러나 OcTree의 최하위 자식 노드들에 속하는 객체의 분할은 이루어졌으나, 상위 부모 노드들의 사용법이 결정되지 않은 상태였다. 따라서 부모 노드들을 활용하기 위해서는 객체들의 병합과 노드 포인트 감소 알고리즘과 그에 대한 룰이 차후 결정되어야 한다.

#### 4.2 경량화 알고리즘 구조

4.1에서 언급된 알고리즘 디자인의 주요 개념을 바탕으로 G3D의 데이터 패키지 구조를 구성하였다. Block-Reference 정보는 Skin, Bone, 그리고 공간인덱싱 대상의 객체들로 구분된다. 대상 객체들은 OcTree의 최하위 자식 노드별로 모아져 각각의 파일로 분리된다. 전체 LOD가 0~M(총 M+1개)까지 존재하고, 이 중 건물 외부용 LOD가 0~N(총 N+1개), 건물내부용 LOD가 N+1~M(총 M-N개)라 할 때, 데이터 패키지 구성은 Figure 7과 같다.

A - Basic Information of Building

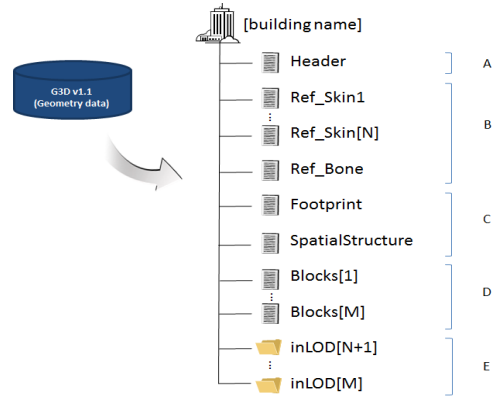


Figure 7. Data Package Structure of G3D

B - Block-Reference data

C - Factor for Camera Position

D - Original Geometry data of Block

E - Spatial Indexing based on LOD

LOD0은 바운딩 박스로 표현할 수 있는 간단한 직육면체로, 관련 기본정보는 헤더에 포함되어 있으므로 따로 생성하지 않는다. 데이터 패키지 구조에서 건물 내부에 해당하는 LOD 정보가 포함된 폴더에는 Figure 7과 같은 파일들로 구성된다. 각 파일은 OcTree의 최하위 자식 노드 각각에 포함된 객체의 Block-Reference 현황을 저장한다. 여기에는 객체의 고유 인덱스, 해당 객체가 참조하고 있는 Block 인덱스, 해당 객체의 변환 매트릭스가 들어있다. 파일명 Ref\_NodeData 뒤에 붙는 숫자는 특정 하위 자식 노드를 가리킨다. 해당 자릿수는 루트 레벨을 제외한 OcTree 레벨 개수와 동일하고 OcTree의 특성상 각 자리는 0에서 7까지의 값을 가질 수 있다.

예를 들어 Figure 8과 같이, Ref\_NodeData362는 루트 노드의 4번째 자식의 7번째 자식의 3번째 자식 노

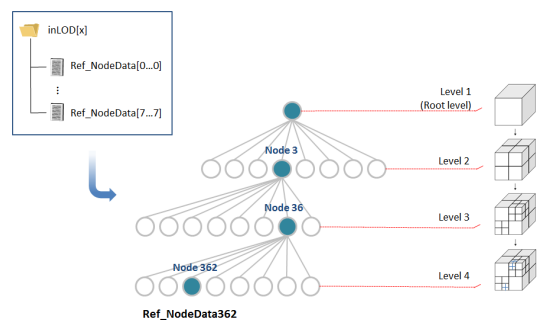


Figure 8. Example of Structure of Nodedata362

Table 1. Description of Data package components

Component	Roll	Contents
Header	Basic information of building	Basic information of building like name, ID, Modeler, etc. from IFC file
Ref_Skin1 - Ref_Skin[N]		Block-Reference data file of object which is located in building skin by LOD level (LOD0~[N])
Ref_Bone	Block-Reference data	Block-Reference data file of object which should be shown ever regardless of spatial indexing when rendering the indoor building
Footprint	Factor for camera position	Expanded foot print data file of building / Use to find whether camera is inside or outside
Spatial Structure		Spatial structure data file of building
Blocks[N+1] - Blocks[M]	Original geometry data of Block	Original geometry data file of object in building by LOD level
inLOD[N+1] - inLOD[M]	Spatial indexing based on LOD	Data folder of spatial indexing by each LOD level which means indoor building (LOD[N+1]~[M])

Table 2. Data format of 'Blocks[LOD level]'

Component name - Blocks[LOD level]			
Total size(bytes) = 4+ 4 x Block Count			
Data type	Name	Length (byte)	Counts by which repeated
Int	Block Count	4	
Int	Block Position	4	Block Count

Table 3. Data format of 'Ref\_NodeData'

Component name - Ref_NodeData[node number]			
Total size(bytes) = 4+(4+4+64) x Object Count			
Data type	Name	Length (byte)	Counts by which repeated
Int	Object Count	4	
Int	Object Index	4	Object Count
Int	Block Index	4	Object Count
Float*16	m00, m01, m02, ..., m33 (components of transform matrix)	64 (4*16)	Object Count

드에 들어있는 객체들이 들어있다. G3D 데이터 패키지를 구성하는 각 요소들에 대한 설명과 실제 G3D에 저장되는 데이터 구성 가운데 Block과 Reference에 해당하는 요소들을 Table 1-Table 4로 각각 정리하였다.

Table 4. Data format of 'Block[LOD level] \_Body sector'

Component name - Block[LOD level] _Body sector					
Total size (bytes)					
$\sum_{i=0}^{N_{sk}-1} \{ 2+4+N_{sk}(i) \times 6+1+1+4+ \sum_{j=0}^{N_{ob}(i)-1} \{ 2+4+ \sum_{k=0}^{N_{ps}(i,j)-1} (3+4+N_{ps}(i,j,k) \times 4)+4+N_{sc}(i,j) \times 8 \} \}$					
Data type	Name	Byte	Counts by which repeated		
Short	Entity Type	2			
Int	Point Count	4			Block Count (Nblk)
Half float *3	X, Y, Z	6 (2*3)			Block Count (Nblk)
Bool	bTransparency	1			Block Count (Nblk)
Bool	bDouble Sided	1			Block Count (Nblk)
Int	Polygon Surface Count	4			Block Count (Nblk)
Short	Color	2		Polygon Surface Count (Npsc)	Block Count (Nblk)
Int	Polygon Count	4		Polygon Surface Count (Npsc)	Block Count (Nblk)
Char *3	NormaX NormaY NormalZ	3	Poly-gon Count (Npc)	Polygon Surface Count (Npsc)	Block Count (Nblk)
Int	Polygon Point Count	4	Poly-gon Count (Npc)	Polygon Surface Count (Npsc)	Block Count (Nblk)
Int	Point Index	4	Poly-gon Point Count (Nppc)	Poly-gon Count (Npc)	Polygon Surface Count (Npsc)
Int	Edge Count	4			Polygon Surface Count (Npsc)
Int*2	Point1 Index, Point2 Index	8 (4*2)		Edge Count (Ne)	Polygon Surface Count (Npsc)

## 5. 알고리즘 테스트 및 분석

### 5.1 경량화 알고리즘 테스트 개요

5장에서는 앞서 정리한 Block-Reference 알고리즘을 검증하기 위하여, 대표적인 3개의 IFC 데이터를 이용하여 실험을 수행하였다. 선정된 IFC 데이터는 형상정보와 속성정보를 모두 포함하는 용량이며, 공간데이터 연계모델인 G3D는 IFC 형태를 가시화하기 위한 기하요소 및 일부 공간에 대한 속성정보의 용량을 포함하고 있다. 이는 공간데이터 연계모델로 변환하는 모듈에서 데이터의 가시화를 위한 형상정보는 파일로 저장되고, 일부 공간정보와 속성정보는 데이터베이스화하여 서비스하도록 설계되었기 때문이다.

실제 공간데이터 연계모델의 핵심 가운데 하나로 대용량 GIS 데이터와 BIM 데이터의 통합 가시화 확보를 위해 본 연구는 Block-Reference 알고리즘 검증으로 형상정보의 가시화 성능을 테스트하였다. 알고리즘이 적용된 전반적인 공간인덱싱 시퀀스는 다음과 같다.

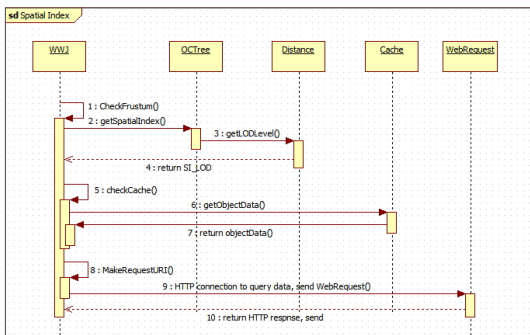


Figure 9. Spatial Indexing Sequence Diagram

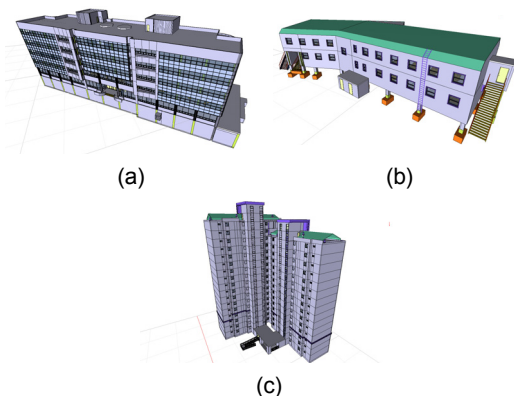


Figure 10. Sample IFC data for Verification

검증에 활용된 IFC 샘플 데이터는 Figure 10과 같다. 이를 바탕으로 Block-Reference 알고리즘을 적용하기 이전의 성능과 알고리즘 적용 후 G3D 성능을 용량과 로딩속도를 중심으로 비교분석하였다.

### 5.2 결과분석

본 테스트에 대한 결과는 Table 5와 같다. 공간데이터 연계모델의 용량의 경우, 고도화 경량화 기술이 적용된 Block-Reference 알고리즘을 적용한 데이터가 그렇지 않은 데이터보다 용량 부분에서 현저히 축소되었음을 확인할 수 있었다. 이러한 용량 축소효과는 데이터를 로딩하는 속도에 영향을 주기 때문에, 실제 공간데이터 연계모델 뷰어에서 검사한 데이터 로딩 속도 또한 알고리즘을 적용한 데이터 파일이 다소 줄어들었음을 확인하였다.

Table 5. Results of the Verification

	(a)	(b)	(c)
File size (Mb)	66.296	28.188	65.339
Size of Non-applied (Mb)	176.054	34.665	74.892
Size of Algorithm applied (Mb)	15.3	4.15	8.41
Speed of Non-applied (sec)	4.889	1.341	3.651
Size of Algorithm applied (sec)	0.64	0.405	0.483

## 6. 결 론

본 논문은 GIS 플랫폼 상의 대용량 BIM 형상정보를 효과적으로 가시화하기 위해 IFC 특성을 고려한 경량화 기법을 제안하고자 하였다. 한국건설기술연구원에서 개발 중인 BIM/GIS 플랫폼에 IFC 데이터 입력 시 속성정보와 형상정보를 구분하여 데이터베이스를 구축한다. 해당 과정에서 형상정보를 공간데이터 연계모델(G3D)로 변환할 때 데이터의 용량 증가로 인해 용량 축소가 필수적으로 요구되는 사항이었다. 이를 해결하고자, 기존에 선행된 다양한 경량화 관련 연구들을 조사·분석하였고 연구환경을 최대한 활용할 수 있는 방안으로서 Block-Reference 기법을 도입하였다. 객체를 최초 타입별로 그룹화하고 기하학적 요소가

동일한 타입을 Block으로 선정, 그 외 복사된 객체들은 Block을 참조하는 형식으로 알고리즘을 설계하였다. 제안된 알고리즘 검증은 3개의 IFC 샘플데이터를 바탕으로 Block-Reference 알고리즘을 적용한 공간데이터 연계모델과 그렇지 않은 기존 모델을 비교분석하였고 검증결과로 데이터 용량이 약 90%, 렌더링 속도는 약 70% 감소되었음을 확인하였다.

향후 건물의 용도별, 디자인 유형별로 추가 알고리즘을 검증하여 경량화 기법을 지속적으로 발전해 나갈 계획이다. 본 논문에서 제안한 렌더링 알고리즘을 적용하면 대용량 GIS 데이터 기반 위에서 대용량 BIM 데이터를 원활하게 가시화할 수 있고 이를 활용하고자 하는 사용자 측면에서도 다양한 서비스 분야를 개척하고 활용할 수 있을 것을 기대한다.

## References

- [1] Fan, H; Meng, L. 2012, A three-step approach of simplifying 3D buildings modeled by CityGML, *International Journal of Geographical Information Science*, 26(6):1091-1107.
- [2] Forberg, A; Mayer, H. 2002, Generalization of 3D building data based on a scale-spaces, *Proc. of Symposium on Geospatial Theory, Processing and Applications, commission IV, WG IV/3*.
- [3] Glander, T; Dollner, J. 2009, Interactive Visualization of Generalized Virtual 3D City Models using Level-of-Abstraction Transitions, *Proc. of Computers, Environment and Urban Systems*.
- [4] Han, S. H. 2014, Implementation of File-referring Octree for Huge 3D Point Clouds, *Journal of the Korean Society of Surveying, Geodesy, Photogrammetry and Cartography*, 32(2):109-115.
- [5] Hwang, J. R; Kang, H. Y; Hong, C. H. 2012, BIM-GIS interoperability and platform development strategy, *Journal of Korea Spatial Information Society*, 99-107.
- [6] Kim, J. E. 2014, Development of the Spatial Indexing Method for the Effective Visualization of BIM data based on GIS, *Journal of the Korea Academia-Industrial cooperation Society*, 15(8): 5333-5341.
- [7] Mao, B; Ban, Y; Harrie, L. 2011, A multiple representation data structure for dynamic visualisation of generalised 3D city models, *ISPRS Journal*

of Photogrammetry and Remote Sensing, 66: 198-208.

- [8] Mayer, H. 2005, Scale-spaces for generalization of 3D buildings, *International Journal of Geographical Information Science*, 19(8-9):975-997.
- [9] Park, J. R; Lee, H. Y. 2013, A Hierarchical User Interface for Large 3D Meshes in Mobile Systems, *Proc. of Korea Computer Graphics Society*, 19(1): 11-20.

---

Received : 2014.12.02

Revised : 2015.1.29

Accepted : 2015.2.27