

# 건물 시설물 관리 관점에서 GIS기반 대용량 BIM 형상 객체 표현을 위한 경량 BIM 형상 포맷 구조 개발에 관한 연구

## A Study on the Lightweight BIM Shape Format(LBSF) Structure Development to Represent the Large Volume BIM Geometry Objects based on GIS as the Viewpoint of the Building Facility Management

강 태 욱\*                      흥 창 희\*\*  
Tae Wook Kang              Chang Hee Hong

**요약** 본 연구에서는 GIS기반에서 대용량 BIM 형상 객체를 효과적으로 표현하기 위한 경량 형상 포맷 구조 개발에 초점을 맞춘다. 최근 건물관리 유스케이스 관점에서 GIS기반에서 수많은 BIM 객체를 관리하고 필요한 정보를 표현하고자 하는 연구가 진행되고 있다. 이를 효과적으로 구현하기 위해서는 IFC와 같은 BIM정보표준모델을 도시차원에서 GIS상에 효과적으로 표현해야 한다. 이를 위해서는 BIM 정보를 포함한 경량화 된 BIM형상 가시화 방법과 포맷 구조를 제안하고 파일럿 테스트를 위해 프로토타입을 구현해 본다. 파일럿 프로세스에서는 연구를 통해 도출된 설계 내용에 따라 3개 지역의 모델 데이터에 대해 일반적인 IFC파일과 경량BIM형상 포맷을 각각 구성한 후, 데이터 용량, 화면 로딩 시간 등의 성능을 비교하였다.

**키워드** : BIM, 경량, 형상, 포맷, 성능

**Abstract** The purpose of this study focuses on the development of the lightweight BIM shape format(LBSF) structure to present the large volume BIM geometry object based on GIS as the viewpoint of the building facility management. Recently, the BIM-based facility management with GIS is researching with the purpose of the urban facility management etc. To implement these use-cases, the BIM geometry objects can be visualized based on urban level with GIS effectively. To do this, the lightweight BIM shape format is designed with the considerations and the prototype for the pilot test is implemented. In the pilot test phase, after developing the IFC file and LBSF file about the model data of the three areas, the performance with the data volume, the screen loading time etc was compared.

**Keywords** : BIM, Lightweight, Shape, Format, Performance

### 1. 서론

최근 GIS와 BIM을 융합해 건물 시설물 관리와 같은 유스케이스를 실현하고자 하는 연구가 진행되고 있다. 이러한 연구에서 문제가 되는 것들 중에 하나는 GIS기반에서 수많은 BIM 객체의 형상정보

를 효과적으로 가시화하고 표현하는 일이다. 보통 BIM분야에서 사용되는 표준모델인 IFC는 구조가 복잡하고 파라메트릭한 정보가 포함되어 있어 그대로 활용하기에는 성능의 문제가 있다. 특히 파일 로딩 시 파일 포맷이 압축되지 않은 텍스트 기반이라 파일 파싱(Parsing) 단계에서 토큰(Token)처리를

<sup>†</sup> This work was researched by the supporting (2013) Development of BIM/GIS Interoperability Open-Platform

\*Tae Wook Kang, Senior Researcher, ICT Lab, Korea Institute of Construction Technology, ktw@kict.re.kr

\*\* Chang Hee Hong, Senior Researcher, ICT Lab, Korea Institute of Construction Technology, chhong@kict.re.kr

포함한 수많은 스트링(String) 처리 연산이 수행된다. IFC내 포함된 수많은 객체에 대한 솔리드(Solid) 파라메트릭 정보를 이용해 형상을 가시화하기 위한 메쉬 처리(Mesh Processing) 단계도 이때 수행된다. 이러한 IFC파일 처리 단계는 결과적으로 오랜 로딩 시간을 필요로 한다.

도시 건물 시설물 관리 차원에서 보면 GIS기반에서 수많은 BIM 객체 형상 가시화 시 데이터가 가지는 성능상의 문제점을 해결할 필요가 있다. 이러한 문제를 해결하면 GIS 데이터와의 효율적 연계가 가능하며 도시 단위의 대용량 데이터에 대한 원활한 가시화 처리 및 인터넷 서비스가 가능하다.

본 연구 범위는 이러한 차원에서 경량화 된 BIM 형상 객체 표현을 위한 방법과 포맷 구조를 제안하고 프로토타입을 구현해 봄으로써 유용성을 확인한다. 이를 위해 형상정보 경량화에 관한 관련 연구를 조사하고 IFC를 처리하기 위해 필요한 단계들을 미리 처리해 놓음으로써 가시화 성능을 개선하고자 한다. 다만, 대용량 BIM 객체 가시화 성능개선 위한 경량화 방법에 초점을 맞추고 있어 파일 포맷에 속성에 관한 LOD(Level Of Detail) 최적화나 공간 인덱싱은 본 연구에서는 포함되지 않았다.

## 2. 연구 방법

본 연구의 방법은 다음 Figure 1과 같이 관련 연구 동향을 우선 조사하고 기존 연구의 한계점과 시사점을 도출한다. BIM 표준 포맷인 IFC 파일의 구조를 분석하여 한계점을 파악하고 GIS기반 도시 시설물 관리차원에서 표현되는 대용량 도시 시설물 객체들의 형상 가시화를 위한 방법과 포맷 구조를 제안한다. 이를 통해 경량 BIM 형상 포맷(LBSF-Lightweight BIM Shape Format)의 구조를 정의하고 LBSF 지원을 위한 클래스와 컴포넌트 구조를 제안한다. 제안된 포맷 구조의 성능을 확인하기 위해 프로그램을 개발하고 파일럿 테스트를 수행한다. 파일럿 테스트에 사용된 3개 샘플을 이용하여 기존 방식과 제안된 방식으로 경량화 된 두 가지 방식을 포함한 세 가지 방식을 비교한다.

## 3. 관련 연구 동향

BIM분야에서 대용량 객체 형상 가시화를 위한

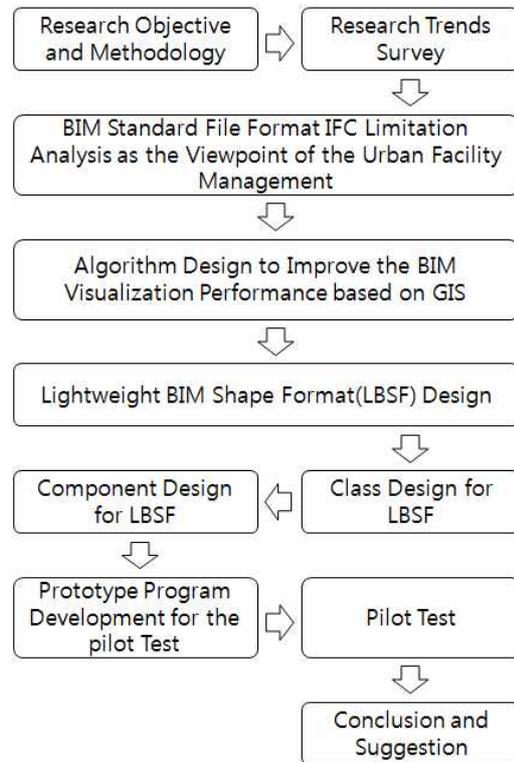


Figure 1. Research Flow

연구는 많지 않다. BIM과 지향하는 방향이 유사한 제조 PLM(Product Lifecycle Management)분야에서는 대용량 객체 형상 정보를 어떻게 효과적으로 가시화할 지에 대한 여러 연구가 진행되었다.

관련해 CAE/CAI에서 생성된 대용량 형상 데이터를 웹 기반에서 협업을 지원하는 연구가 있었으며 생성된 데이터를 경량화하여 가시화 표현을 통해 웹 기반에서 협업할 수 있는 방법론과 자료구조를 연구하였다[1]. 여기서 적용한 방법은 형상 메쉬에서 해석에 크게 영향을 미치지 않는 부분은 특정 조건에 의해 메쉬를 간략화함으로써 경량화하는 방법을 사용하였다. 이 연구는 해석 결과로 출력되는 메쉬에 대해서만 메쉬 간략화를 수행한 연구로 건물시설물 관리차원의 활용 목적과는 차이가 있다.

이외에 JT포맷과 JT Toolkit을 이용해 해석데이터를 경량화하여 데이터의 원활한 공유 및 공유를 위한 연구가 있었으며[5]. 가상 공장 시뮬레이션을 위해 3차원 데이터 경량화를 위해 은면체기 시 그래픽 GPU를 활용하고 메쉬 간략화 기법을 적용하는 등에 관한 연구가 있었다[3].

PLM분야에서도 국제 표준으로 IGES, STEP등이 정보 교환에 초점이 맞추어져 있어 크기가 크고 비

효율적이라는 지적이 있다[7]. 이런 이유로 JT, U3D, 3D-XML과 같은 상용 PLM에 활용되는 경량 가시화 데이터 포맷이 활용되고 있다.

상용 PLM에서 활용되는 포맷을 살펴보면 JT는 1998년 형상 가시화의 효과적인 표현을 위해 HP와 Engineering Animation Inc. 참여로 Jupiter Project란 이름으로 진행되었으며[9], B-Rep구조와 속성정보를 포함하고 있고 데이터 압축 및 LOD기술을 지원하나 포맷 구조가 복잡하며 이를 실질적으로 활용하기 위해서는 라이선스와 개발 도구를 구입해야하는 문제가 있다. U3D는 2004년 인텔을 중심으로 Adobe 등 24개 회사가 참여한 3D산업 포럼에서 발표한 표준 포맷으로 데이터 스트리밍을 지원하여 다운로드 동안 점진적 뷰잉을 지원하나 메쉬 분할에 대한 추가연산으로 인한 속도저하가 지적되고 있다[10]. 3D-XML은 Dassault사에서 제안하였고 XVL이라는 정점, 위상, 부가 정보만 관리하고 있어 구조가 간단하다[6].

관련 연구들을 보았을 때 대부분 PLM분야에서 관련연구가 이루어져 건물시설물 관리 차원에서 대용량 BIM 형상 데이터를 효과적으로 가시화하는 방법과 구조 관련 연구는 부족하고 라이선스 문제 등으로 인해 직접적으로 활용하기 어려운 한계가 있다.

#### 4. 건물시설물 관리 차원에서 BIM표준 포맷 IFC파일 활용 시 한계점 및 고려사항

최근 공식 발표된 BIM표준 포맷인 IFC4는 건축/건설정보교환 표준화기구인 buildingSMART협회에서 IFC 2x3에서 여러 문제점을 개선한 버전으로 2013년 3월에 발표되었다. IFC는 객체지향적으로 설계된 구조로 현재 buildingSMART에서 지속적으로 건축분야 정보교환을 위한 표준화활동을 하고 있다. IFC4는 모델링 시 정보 구조의 모호성을 제거하려 노력하였으며, NURBS지원을 통한 비정형 모델 표현, 좌표계 지원을 통한 GIS와 연계, 4D/5D 정보 모델 개선 등이 있었다. 데이터 경량화와 관련해서는 ifcXML포맷에서 사용된 XML태그 수를 줄이고 개별적으로 정의된 데이터 노드를 그룹으로 병합해 데이터 리스트 형태로 간략화하여 14%경량화 하였음을 밝히고 있다[8].

IFC4는 데이터 경량화를 통한 가시화 성능 개선

을 위한 노력이 있었으나 근본적으로 파라메트릭 모델 표현과 정보 교환에 초점을 맞추고 있어 가시화에 필요한 정보들을 계산해 줘야 하는 문제가 있다. 예를 들어 수학적 파라미터로 표현되어 있는 벽체나 지붕과 같은 경우 가시화하기 위해서는 별도의 메쉬 처리 연산이 필요하다.

다음 Figure 2는 IFC파일의 객체들을 추출하고 각 객체들의 형상 정보를 가시화하는 방법을 분석한 알고리즘이다.

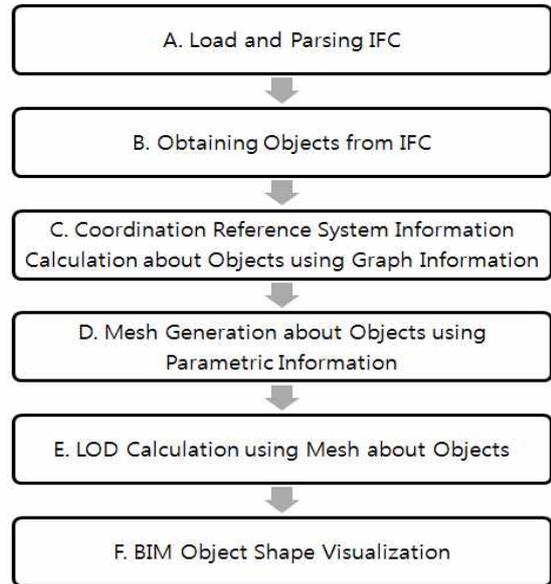


Figure 2. IFC Model Visualization Algorithm

IFC파일은 형상에 관한 정보를 표현할 때 기본적으로 OpenGL등에서 활용하는 그래프 기반의 계층 구조로 지역좌표계(Local Coordination Reference System)를 표현한다. 이런 이유로 IFC를 파싱하고 객체를 얻은 후에 지역좌표계를 전역좌표계(World Coordination Reference System)로 계산해 변환해야 한다. 이후 각 객체 형상에 수학적으로 표현된 파라메트릭 정보를 이용하여 메쉬 처리(Mesh Processing)를 한 후 적절한 가시화 성능을 얻기 위해 LOD를 계산한다. LOD는 3장에서 언급한 메쉬 간소화 기법을 사용하고 간소화 기법의 대부분은 메쉬 간소화 목적에 따른 메쉬를 구성하는 정점, 모서리, 면을 제거하는 방법에 대한 것들이다. 이후 얻어진 LOD의 메쉬를 OpenGL과 같은 렌더링 엔진을 이용해 화면에 렌더링하여 객체 형상을 가시화한다.

이런 전 과정에 계산 기하학적인 수많은 연산들이 포함되고 있으며, 이런 연산들은 건물시설물 관리 관점에서 보면 굳이 관리할 필요가 없는 정보와 계산들이 많다는 것을 알 수 있다.

특히, 관리와 운영 차원에서 보면 다음의 정보들은 불필요하다.

1. 객체별 지역 좌표계
2. 객체별 형상정보 생성에 관련된 파라메트릭 정보

이와 관련된 단계인 C, D, E는 형상 모델링하는 목적이 아니라면 불필요한 처리 단계이다. 또한, IFC를 파싱하는 단계도 수많은 문자열 처리 및 구문 해석처리가 포함되므로 이 부분도 미리 처리할 수 있다면 GIS기반의 수많은 건물시설물 BIM객체들을 로드하는 데 속도를 개선할 수 있다. 이를 고려하여 IFC 가시화 단계 중 미리 처리할 수 있는 형상 정보들을 생성하기 위한 알고리즘을 디자인하고 이 정보들을 담기 위한 포맷을 디자인 한 후 컴포넌트와 클래스 구조를 제안한다.

## 5. 경량 BIM객체 형상 모델 디자인

### 5.1 알고리즘 디자인

4장에서 도출된 고려사항을 바탕으로 파라메트릭 기반의 BIM 형상 데이터를 표면정보(Exterior Surface) 형태로 변환하기 위해 다음 Figure 3과 같이 알고리즘을 디자인하였다.

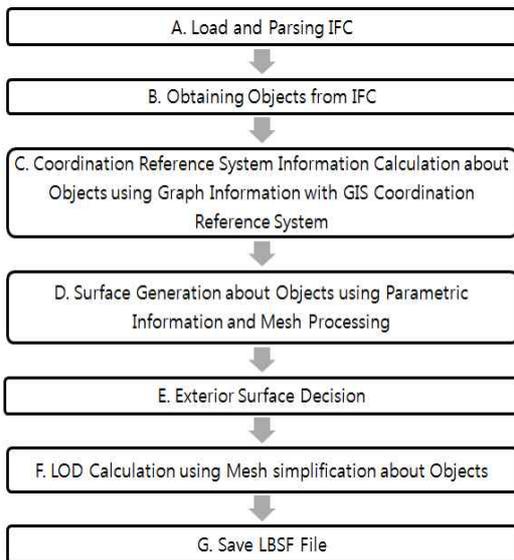


Figure 3. File Format Conversion Algorithm From IFC to LBSF

건물시설물 객체 관리차원에서 불필요한 연산인 전역좌표계 변환 및 모델링 정보 생성하는 부분들을 미리 계산하고 메쉬 간소화기법을 이용해 LOD를 계산한다.

대용량 데이터의 원활 서비스를 위해 최외곽 표면정보만을 추출해 가시화 성능 개선과 관련된 CityGML의 LOD 처리 개념을 고려하였다. 그리고 실제 GIS상에 BIM객체들을 가시화할 경우 IFC포맷에서 변환된 LBSF파일을 로딩해 사용하도록 한다.

최외곽 표면정보 추출 알고리즘은 간단하게 다음 Figure 4와 같이 생성된 BIM 형상에서 형상을 구성하고 있는 각 표면정보에 대해 카메라를 형상 중심점을 바라보게 하고 BIM 형상 주변 상하좌우에 배치시켜 보이는 표면정보를 검사하는 방식으로 진행한다. 다만, 이 방식은 곡면이나 복잡한 형태에서 카메라에 보이지 않은 표면정보가 있을 경우 제대로 최외곽 표면정보를 추출하기 어려운 한계가 있어 향후 개선이 필요하다.

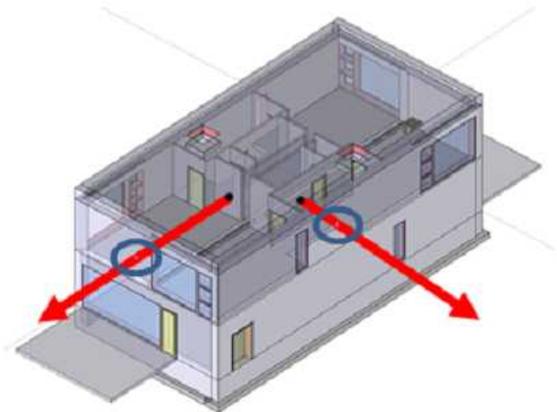


Figure 4. Exterior Surface Extraction Method

### 5.2 파일 포맷 디자인

파일 포맷은 다음과 같은 요구사항을 만족하도록 디자인되었다.

1. 가시화 성능: 가시화시 기하학적인 계산처리를 최대한 줄여야 함. 이를 위해 미리 계산된 표면 정보를 사용하고 LOD를 관리할 수 있어야 함.
2. 최소화: 파일 크기를 최소화할 수 있어야 함.
3. 중복성 배제: 파일 포맷에서 반복적으로 나타나는 데이터는 분리하고 인덱스로 참조함.
4. 압축성 고려: 파일 포맷에서 성격이 유사한 요

건물 시설물 관리 관점에서 GIS기반 대용량 BIM 형상 객체 표현을 위한 경량 BIM 형상 포맷 구조 개발에 관한 연구

소들은 그룹으로 관리하여 그룹 내에서 압축과 해제가 손쉽게 일어날 수 있도록 함.

파일 포맷의 구조는 다음 Figure 5와 같이 크게 4가지 부분으로 구성된다.

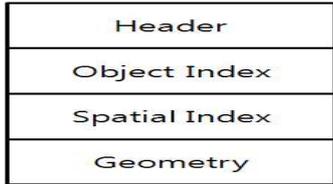


Figure 5. File Format Overall Structure

헤더(Header)는 파일 버전, 객체 위치 및 방향, 객체 개수와 같은 정보를 포함한다. 객체 인덱스(Object Index)는 가시화 성능을 고려하기 위해 LOD 청크(Chunk) 리스트를 관리한다. 공간 인덱스(Spatial Index)는 가시화되는 영역만 표현하기 위해 옥트리(Octree)로 옥트리 노드(Node)가 객체를 관리하게끔 한다. 형상(Geometry)는 객체들을 표현하는 표면정보들의 실제 좌표 리스트로 구성된다.

헤더의 자세한 구조는 다음과 같다.

Table 1. Header Structure

Data Type	Name	Length (Byte)	Description
char	Signature	10	File Format Signature
char	Version	3	Version
string	File-name	255	File name
integer	Object-Count	4	Object count
integer	LOD-Count	4	LOD count
bool	Is-Exterior	1	Exterior Surface Flag
vector	Location	20 = 8+8+4	Object location
vector	Orientation	12=4x3	Object orientation
vector	Scale	12=4x3	Object scale
rect3d	BoundingBox	48=8x6	Object Maximum Bounding Box

객체 인덱스 구조는 다음과 같다. 확장성을 고려해 객체별로 LOD수준을 어플리케이션 목적에 따라 가변적으로 처리할 수 있도록 한다.

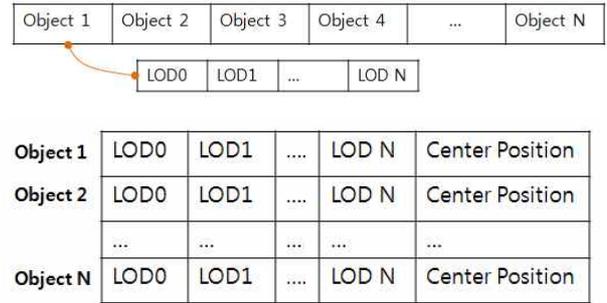


Figure 6. Object and LOD Structure

LOD의 일반 정보를 관리하는 헤더 구조는 다음 Table 2와 같이 LOD 수준, 형상 정보 시작 위치, 형상 정보 길이를 관리한다.

Table 2. LOD General Information Structure

Data Type	Name	Length (Byte)	Description
byte	LOD-Level	1	Level Index of LOD
long	Start-Position	4	Geometry Start Position
long	Length	4	Geometry Length

LOD를 포함하고 있는 객체별 정보 구조는 다음 Table 3과 같다. 객체의 시작 위치, 끝 위치와 LOD 일반 정보가 개수만큼 가변적으로 포함되어 있다.

Table 3. Object Information Structure

Data Type	Name	Length (Byte)	Description
string	GUID	22	Object GUID
long	Start-Position	4	Start Position in File Stream
long	EndPosition	4	End Position in File Stream
LOD	LOD	variables	LOD data stream

공간 인덱싱 구조는 본 연구에서는 구현되지 않았지만 차후 확장성을 위해서 관련 정보가 들어갈 영역을 포맷 구조에 할당만 하였다.

형상(Geometry)부분의 포맷 구조는 다음 Table 4

와 같으며 형상의 유일ID와 LOD별 기하형상을 표면정보 형태로 표현하고 있는 정보들의 리스트로 구성되고 표면정보는 정점, 모서리, 면으로 구성된다.

Table 4. Geometry Information Structure

Data Type	Name	Length (Byte)	Description
string	GUID	22	Object GUID
binary	LOD-List	variables	LOD Geometry Information List

### 5.3 클래스 디자인

클래스 구조는 개발된 포맷을 고려해 다음 Figure 7과 같이 객체지향적으로 구조화하였다.

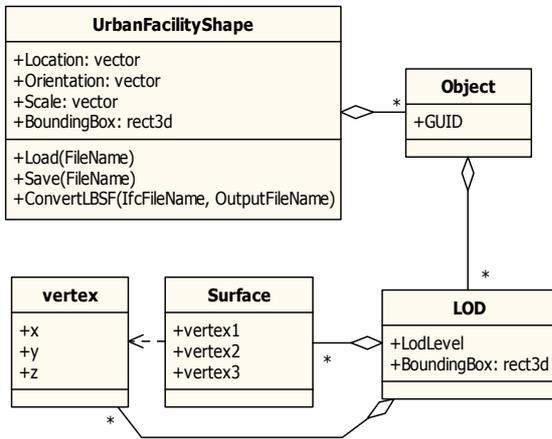


Figure 7. LBSF Class Structure (UML)

### 5.4 컴포넌트 구조 디자인

컴포넌트 구조는 다음 Figure 8과 같다.

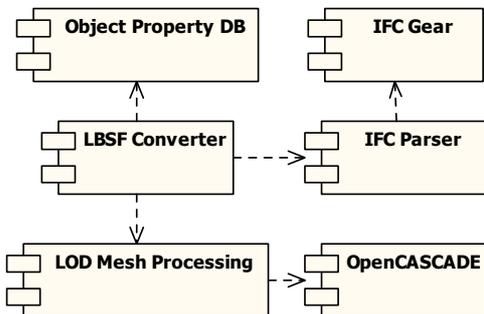


Figure 8. LBSF Component Structure

각 컴포넌트의 주요 역할은 다음과 같다.

**Object Property DB:** 객체별 속성정보를 별도로 관리하고 있으며 객체 GUID로 유일하게 구분되고 있다. 이 객체 GUID는 BIM 객체의 GUID와 동일한 값을 가지며 LBSF내 형상 모델과 속성이 연계될 경우 활용된다.

**IFC Parser:** IFC 파일 포맷을 읽어 파싱하는 컴포넌트이며 오픈소스인 IFC Gear의 기능을 활용한다.

**LBSF Converter:** 파싱한 각 객체별로 형상을 언어 LOD별로 메쉬 처리하고 메쉬를 단순화 처리하는 컴포넌트이다.

**LOD Mesh Processing:** LOD별로 메쉬 처리하는 컴포넌트이며 오픈 소스인 OpenCASCADE의 일부 기능을 활용한다.

## 6. 비교 검증

제안된 경량화 방법을 통해 가시화 성능을 비교하기 위해 파일 로딩 시간, 데이터 용량, 메모리 사용량에 대하여 제안된 방법인 표면정보 변환 방법과 최외곽 표면정보 추출 방법 그리고 기존 IFC 포맷을 사용하는 방법 세가지를 비교하였다.

Table 5. LOD General Information Structure

Data	Name	Shape Type
Sample #1	Cheongun University Main and Academic Information Center	
Sample #2	Family Restaurants	
Sample #3	Sungkyunkwan University dormitory building	

다음 Table 6은 성능 비교 검증에 사용된 샘플의 데이터 특성이다. 참고로 현재 BIM으로 진행되는 프로젝트는 다수 있으나 샘플로 사용하기에는 현실적으로 접근 상 어려움이 있어 이미 모델링이 되어 있으며 학술 연구용으로 활용이 허락된 규모가 있는 샘플을 선정해 사용하였음을 밝힌다.

Table 6. Used Samples Characteristics

Type	Vertex Count	Data Volume
Sample #1	15,454,362 points	301,844 KB
Sample #2	19,800,485 points	386,729 KB
Sample #3	643,279,768 points	17,587,733 KB

각 샘플별로 제안된 방식을 대용량 BIM 형상 가시화를 위해 중요하다고 생각되는 로딩 시간, 메모리 사용량, FPS(Frames Per Second) 지표로 테스트하였다. 성능 측정 지표로 사용된 IFC 파일을 로딩하는 시간에는 파싱 처리 단계와 형상을 매쉬 처리하는 단계가 포함되어 있는 데 경량 BIM 형상 포맷에는 이미 처리되어 있으므로 속도 개선을 기대할 수 있다. 메모리 사용량은 성능 개선과 관련해 특별한 알고리즘이나 데이터 포맷을 사용할 때 영향을 받는다. 제시한 방법이 메모리 사용량에 어떤 영향을 미치는 지 알아보기 위해 메모리 사용량을 확인하였다. FPS는 응답성 관련된 것으로 수치가 낮을수록 형상 정보를 관찰하는 것에 어려움을 느낄 수 있다. 일반적으로 FPS 수치는 높을수록 좋다.

다음 Figure 9는 프로토타입 프로그램 구동 시 화면이다.



Figure 9. Prototype Program Screen

각 샘플별로 테스트를 수행한 결과는 Table 7과 같다.

정량적 성능평가 결과에 의하면 LBSF 포맷은 데이터 로딩 시간은 샘플#2의 경우 95%감소하는 것으로 나타났으며 이는 LBSF포맷이 파싱 처리 단계와 매쉬 처리 단계를 미리 처리하여 이 부분에서 성능 개선이 있었다고 판단된다. 메모리 사용량은 샘플#1의 경우 최대 64%감소하였고, 모델 데이터 용량과

Table 7. Test Results about Samples (Data Volume, Loading Time, Memory Usage, FPS)

Format and Performance		Sample #1	Sample #2	Sample #3
IFC	Data Volume (MB)	67	26	12
	Loading Time (Second)	22.05	217.22	5.99
	Memory (MB)	499	1,029	156
	FPS	16.44	35.71	33.78
LBSF	Data Volume (MB)	32	213	11
	Loading Time (Second)	1.28	8.69	0.44
	Memory (MB)	177	890	75
	FPS	16.44	33.67	35.58
LBSF with the exterior surface	Data Volume (MB)	7	66	4
	Loading Time (Second)	0.30	2.59	0.20
	Memory (MB)	56	286	43
	FPS	42.73	58.14	64.1

FPS에 대해서는 큰 차이가 없었다. 이는 두 포맷 모두 렌더링해야 하는 표면정보의 개수에는 크게 변함이 없기 때문으로 판단된다. 결론적으로 표면정보를 미리 계산해 놓은 LBSF포맷은 데이터 로딩 시간이 크게 개선된 것을 알 수 있다.

LOD3 이하에 해당하는 최외곽 표면정보를 처리한 LBSF의 LOD 모델 데이터 용량은 샘플#1의 경우 89% 정도로 감소하는 것으로 나타났으며, 데이터 로딩 시간은 샘플#2의 경우 98%, 샘플#1의 경우 메모리 사용량은 약 88% 정도로 감소하였으며, FPS는 샘플#3의 경우 약 1.9배 개선된 것으로 나타났다. 렌더링해야 하는 표면정보 개수가 줄어들었으므로 전반적으로 성능이 크게 개선되었다.

다음 Figure 10-13과 같이 각 샘플에 대한 포맷 성능을 그래프로 분석해 보면 전반적으로, IFC 포맷의 BIM 데이터를 LBSF 모델로 변환하는 경우 로딩 시간, 메모리 사용량이 급격히 감소하고 FPS가 향상되는 것을 알 수 있으며 이는 시험 전에 기대한 결과와 같다.

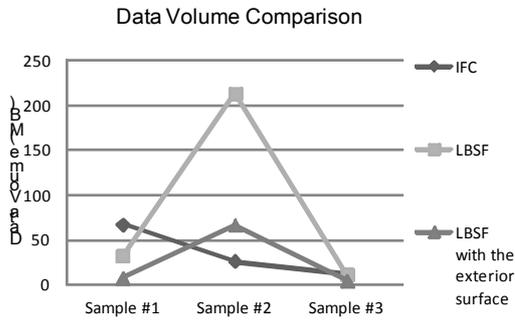


Figure 10. Samples Data Volume Comparison about each format

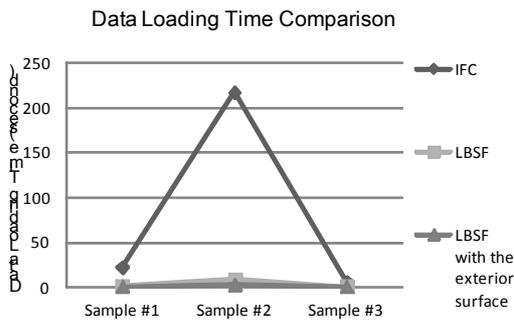


Figure 11. Samples Data Loading Time Comparison about each format

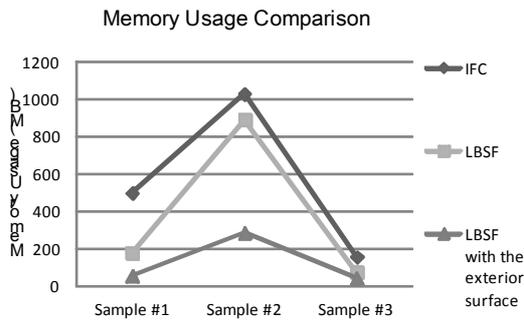


Figure 12. Samples Memory Usage Comparison about each format

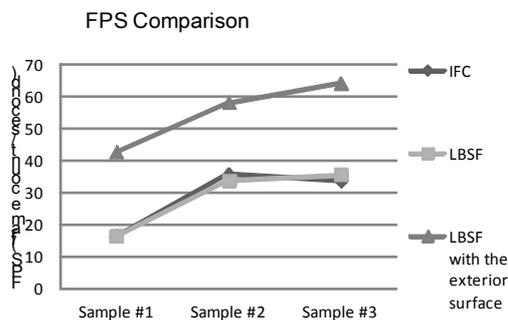


Figure 13. Samples FPS Comparison about each format

## 7. 결론

본 연구는 건물 시설물 관리 차원에서 필요한 GIS기반에서 수많은 BIM 객체 형상 가시화 시 데이터가 가지는 성능상의 문제점을 해결하기 위해 경량화 된 BIM형상 객체 표현을 위한 방법과 포맷 구조를 제안하고 프로토타입을 구현하였으며 3개의 샘플과 4개의 지표를 통해 성능 개선 관점에서 유용성을 확인하였다

비교 검증 결과에서 보는 바와 같이 건물시설물 관리 관점에서는 모델링에 필요한 모든 정보가 포함된 IFC포맷을 활용한다면 GIS과 같은 대규모 지형에 표현되는 수많은 BIM형상을 효과적으로 표현하기 어렵다. 이러한 경우 시설물 관리 관점에서 필요한 형상과 속성만 추출한 포맷을 활용하는 것이 효과적이며 이와 관련해 제안한 LBSF포맷 성능이 IFC포맷을 그대로 활용할 때 보다 뛰어난 것을 알 수 있다. 특히 GIS관점에서 모든 BIM객체를 렌더링할 때는 최외곽 표면정보만 가지고 있는 LBSF포맷이 성능을 고려해 볼 때 유리하므로 사용자가 보는 관점과 LOD에 따라 전략적으로 각 포맷을 활용할 필요가 있다.

향후 계획은 가시화 성능개선 위한 공간 인덱싱을 함께 고려하고 최외곽 표면정보를 처리하는 메쉬 프로세싱부분을 개선하여 성능을 높이고 데이터 압축 기법 및 가상메모리 관리기법을 연구해 GIS기반 대용량 BIM객체 분산 처리 기법을 개발할 계획이다.

## References

- [1] Chang, K. J. 2011, A study on the expression of lightweight-visualization about CAE/CAI data and collaboration in a network environment, Master Thesis, Chungang University.
- [2] Hwang, J. R; Kang, T. W; Hong, C. H. A Study on The Correlation Analysis Between IFC and CityGML for Efficient Utilization of Construction Data and GIS Data, Korea Spatial Information Society, 20(5):49-56.
- [3] Kwak, J. G. 2011, Theoretical study on construction of 3D data for control logic validation, Doctoral Thesis, Ajou University.

- [4] Kang, T. W; Hong, C. H; Hwang, J. R; Choi, H. S. 2012, The External BIM Reference Model Suggestion for Interoperability Between BIM and GIS, Korea Spatial Information Society, 20(5):91-98.
- [5] Lee, O. L. 2009, Lightening CAE Analysis Data using JT, Master Thesis, Hanyang University.
- [6] Park, K. H. 2008, Automatic 3D PDF Generation on a 3D Viewer System, Master Thesis, Chungang University.
- [7] Reed, K; Harrod, D; Conroy, W. 1990, the Initial Graphics Exchanges Specification(IGES) Version 5.0. NIST.
- [8] Thomas, L. 2013, IFC4 - the new building SMART Standard.
- [9] UGS, 2006, JT File Format Reference Version 8.1.
- [10] Wakita, A; Yajima, M; Harada, T; Toriya, H; Chiyokura, H. XVL, A Compact and Qualified 3D Representation.

---

논문접수 : 2013.05.04

수정일 : 2013.06.26

심사완료 : 2013.06.27