Development of Lightweight BIM Shape Format Structure to Represent Large Volume Geometry Objects Using GIS with Facility Management

T.W. Kang[a] and C.H. Hong[b]

[a]ICT Lab, Korea Institute of Construction Technology, Republic of Korea
[b]ICT Lab, Korea Institute of Construction Technology, Republic of Korea
E-mail: laputa99999@gmail.com, chhong@kict.re.kr

Abstract -

This study focuses on the development of a lightweight building information management (BIM) shape format (LBSF) structure to represent large-volume BIM geometry objects using the geographical information system (GIS) with building facility management. Recently, BIM-based facility management with GIS has been researched with regard to urban facility management. To implement this use case, the BIM geometry objects are required to be effectively visualized on the urban level using GIS. Therefore, a lightweight BIM shape format is designed with these considerations, and the prototype for the pilot test is implemented. In the pilot test phase, after developing the Industry Foundation Classes (IFC) file and LBSF file for the model data of the three areas, performance comparison with regard to the data volume and screen loading time was made.

Keywords -

BIM; Lightweight; Shape; Format; Performance

1 Introduction

Recently, studies have been conducted on the integration of the geographical information system (GIS) and the building information management (BIM) to achieve use cases such as building facility management. One of the challenges for these studies is effective visualization and representation of the geometric information of multiple BIM objects using GIS. IFC, which is the standard model generally used in the BIM field, has a complex structure and possesses parametric information; therefore, it experiences performance-related issues when used. In particular, as IFC is text-based with an uncompressed file format, processing of multiple strings and tokens is performed during the file parsing stage when the file is loaded. Consequently, to visualize the geometry of an object, mesh processing is also performed using the solid parametric information on a variety of objects included in IFC. Therefore, these IFC file processing stages require a long loading time.

With regard to urban building facility management, the performance-related issues that occur when the geometries of numerous BIM objects are visualized using GIS must to be solved. For example, when rendering or selecting a BIM object, and when the performance is too low for the user to operate the system, it is difficult to perform tasks such as confirmation of the BIM object data. Thus, to effectively visualize the geometric data of large-volume BIM objects, weight reduction the large-volume geometric information is required. It is possible to efficiently connect the GIS data, visualize the large-volume data of the city unit, and provide Internet services on the data when these issues are solved.

Therefore, this study proposes a method and a format structure to represent lightweight BIM geometry objects and realizes the prototype to confirm the effectiveness of visualization of the geometric information. Accordingly, studies on the weight reduction of geometric information are investigated, and necessary steps for IFC processing are performed in advance to improve the visualization performance. However, as this study focuses on the method for weight reduction of large-volume BIM objects, the level of detail (LOD) maximization of file format characteristics and spatial indexing have not been included.

2 Study Method

In comparison to the study method, as shown in Figure 1, the current trend of the relevant studies is investigated, and the limitations and implications of the previous studies are derived. The limitations are identified by analyzing the structure of the IFC files and standard BIM format, and a method and a format structure are proposed to visualize the geometry of large-volume urban facility objects that are represented by GIS-based urban facility management. Subsequently, the structure of lightweight BIM shape format (LBSF) is defined, and the class and component structure for LBSF assistance are proposed. To confirm the performance of the proposed format structure, a program is developed, and a pilot test is conducted.

By using three samples in the pilot test, we compare the three methods including the conventional method and the two proposed methods in which weight reduction of geometric information has been performed. The verification methods by sample are tested with regard to the loading time, memory usage, and frames per second (FPS) indexes.

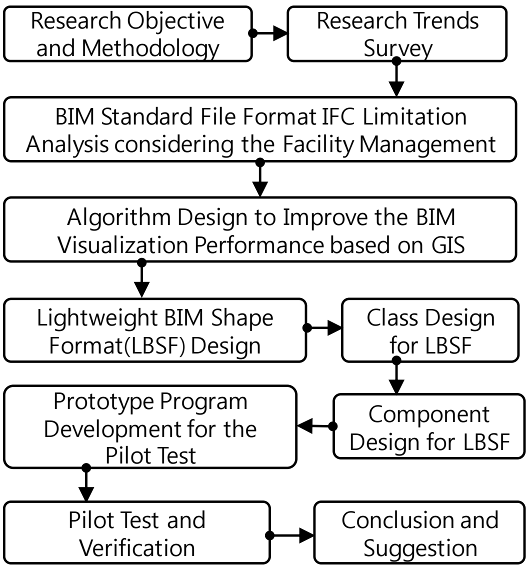


Figure 1. Research flow

3 Current Status of Relevant Studies

Few studies on the visualization of large-volume object shapes have been conducted in the BIM field. In the product lifecycle management (PLM) field, which is similar to the BIM field, the method to effectively visualize the geometric information on large-volume objects has been widely studied.

Studies on promoting Web-based cooperation using large-volume shape data created in CAE/CAI were also carried out, and the methodology and data structure to reduce and visualize the created data for Web-based cooperation were investigated [1]. In the applied method, the mesh was simplified under certain conditions to reduce the weight of the parts that had an insignificant influence on the shape mesh analysis. As mesh simplification was only performed on the mesh that was output as the analytic result, the objectives of these studies are different from that of the building facility management.

Except the abovementioned studies, those on smooth data exchange through the weight reduction of analytic data using the JT format and the JT Toolkit were conducted [5]. Moreover, studies on using the Graphic Process Unit (GPU) and applying the mesh simplification method to eliminate the hidden surfaces for virtual factory simulation and three-dimensional (3D) data weight reduction were conducted. In the study on the 3D visualization using CityGML [11], the model shape was visualized through WebGL, which supports the 3D acceleration function of the GPU. However, the improvement of the visualization performance through the data format was not considered in that study. In another study related to WebGL [12], a lightweight structure considering data compression and data streaming was proposed, whereas LOD processing required to visualize BIM object shapes using GIS was not considered.

Further, in the PLM field, IGES and STEP, the international standards focusing on information exchange, were criticized for their significant size and inefficiency [7]. Therefore, lightweight visualization data formats such as JT, U3D, and 3D-XML, which are used for the common PLM, have been used.

With regard to the formats used for common PLM, JT was developed through the Jupiter Project by HP and the Engineering Animation Inc., for efficient shape visualization, in 1998 [9]. However, while providing data compression and LOD technologies, JT includes the B-Rep structure, attribute data, and complex format structure. Therefore, practical application of the format requires purchase of license and a development tool. U3D is the standard format announced in the 3D industry forum in which 24 companies on the basis of Intel participated in 2004. Although gradual viewing is provided through data streaming in the download process, speed degradation due to additional operations for mesh separation was pointed out [10]. 3D-XML proposed by Dassault has a simple structure as it only manages a vertex, a phase, and additional information.

According to the relevant studies mainly conducted in the PLM field, the method and structure required to effectively visualize large-volume BIM shape data in terms of building facility management have been rarely researched upon, and it is difficult to directly use the solutions of previous studies owing to various issues such as license.

4 Limitations and Considerations in the Case of Using the IFC Files and the BIM Standard Format with Facility Management

In March 2013, the buildingSMART alliance, a standards organization whose objective is to improve the exchange of information on architecture and construction, officially announced IFC4, the BIM standard format. It is the upgraded version of IFC 2x3, and various problems of the previous version have been solved. IFC has an object-oriented structure, and the buildingSMART alliance is continuously promoting standardization activities to exchange information on construction. The objective of the IFC4 modeling is to eliminate the ambiguity of the information structure, and the performance was enhanced, which included irregular model representation through the NURBS support, GIS connection through the coordinate system support, and improvement of the 4D/5D information models. With regard to data weight reduction, a reduction of 14% was achieved by decreasing the number of XML tags used in the ifcXML format and merging individually defined data nodes into groups in order to simplify them in the form of the data list [8].

We have attempted to improve the visualization performance of IFC4 through data weight reduction. However, the information required for visualization should be calculated because the focus is on representing parametric models and exchanging information. For example, additional mesh processing operations are required to visualize the walls and roofs that are represented as mathematical parameters.

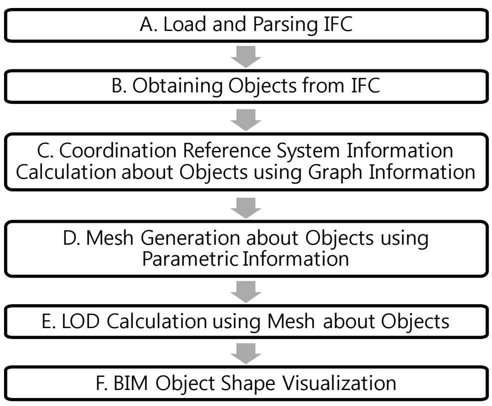


Figure 2. IFC model visualization algorithm

Figure 2 shows the algorithm used to analyze the method to extract objects of the IFC files and visualize the geometric information on each object.

For expressing the geometric information, the IFC files represent the local coordination reference system in the graph-based classification structure mainly used in Open GL. Therefore, after obtaining the objects through IFC parsing, the local coordination reference system should be converted into the world coordination reference system for calculation. Subsequently, by using the parametric information mathematically expressed in each object shape, mesh processing is performed, and LOD is calculated to estimate the appropriate visualization performance. For LOD, the mesh simplification method mentioned in Chapter 3 is used. Most simplification methods focus on removing the peak, corner, and side, which consist of the mesh, as per the requirement. Further, the mesh of the obtained LOD is rendered on a screen using a rendering engine such as OpenGL to visualize the object shape.

In the entire process, a significant number of geometric calculations are involved, and these operations comprise significant information and calculations that need not be considered for building facility management.

The following information is particularly unnecessary from the perspective of management and operation:

1. The local coordination reference system of the object
2. Parametric information for mesh generation of the object

The relevant C, D, and E stages are unnecessary unless shape modeling is required. Moreover, the stage of parsing IFC includes the processing of numerous character strings and syntactic analysis. When these parts are processed in advance, the speed of loading GIS-based BIM objects of various building facilities is increased. Accordingly, the algorithm that creates geometric information to be processed in advance in the IFC visualization stage and the format that includes this information are designed. Furthermore, the LBSF component and class structure are proposed.

5 Design of the Lightweight BIM Object Shape Model

5.1 Algorithm Design

On the basis of the considerations derived in Chapter 4, the algorithm that converts the parametric BIM shape data into an exterior surface has been designed as shown in Figure 3.

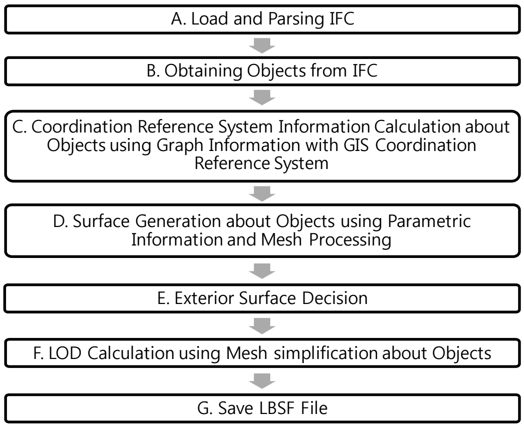


Figure 3. File format conversion algorithm from IFC

The conversion of the world coordination reference system and the creation of modeling information, which are unnecessary operations in terms of building facility object management, are calculated in advance, and the LOD is also obtained using the mesh simplification method.

To create a mesh of certain shape, the process of transforming a curved surface with a curvature into a mesh is required. The accuracy of the mesh is expressed by the number of segments in which the curves are approximated as straight lines as follows:

$$S_N = f(r)$$

The f() function determines S_N by applying the following rules:

$$\begin{aligned}
&\text{if } r < 0.03, \text{then } S_N = 4 \\
&\text{else if } r < 0.05, \text{then } S_N = 6 \\
&\text{else if } r < 0.1, \text{then } S_N = 8 \\
&\text{else if } r < 0.5, \text{then } S_N = 10 \\
&\text{else } S_N = 36 \times \frac{a}{\pi \times 2}
\end{aligned}$$

The accuracy of the curved surface expressed by the mesh varies according to the determined S_N. However, the issue of shape accuracy is not considered in the research range of this study because the objective of this study is to improve the visualization performance in terms of urban facility management.

For smooth services of large-volume data, the concept of LOD processing by CityGML, which is related to the improvement of visualization performance, is considered by extracting the exterior surface information. When the BIM objects are visualized in the practical GIS, the loaded LBSF files converted from the IFC format are used.

As shown in Figure 4, the algorithm for extracting the exterior surface data is employed by placing cameras to focus on the central point of the BIM shape for data on each surface and then examining the surface data located around the BIM shape. However, this method cannot properly extract the exterior surface data when certain surface information is not observed by the camera in curved or complex surfaces, and therefore, it is necessary to improve this method.

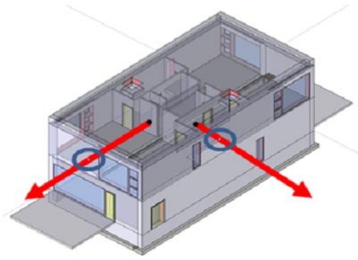


Figure 4. Exterior surface extraction method

5.2 Design of the File Format

The file format is designed to satisfy the following requirements:

1. Visualization performance: geometric calculations should be minimized during visualization. For this, the surface information calculated in advance is used, and the file format should manage LOD.
2. Minimization: the file size should be minimized.
3. Elimination of redundancy: the data repeatedly shown in the file format should be separated and referred to as the index
4. Compressibility: factors with similar properties in the file format are managed as groups to easily facilitate compression and decompression within the groups.

The structure of the file format comprises four main parts as shown in Figure 5.

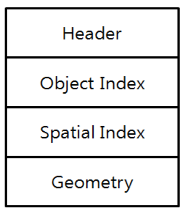


Figure 5. File format of the overall structure

The header includes information such as the file version, location, direction, and number of objects. The object index manages the LOD chunk list to consider the visualization performance. The spatial index enables the octree nodes to manage objects only to represent the visualized areas. The geometry comprises a list of actual coordinates of the surface data that represent the objects. The detailed structure of the header is as follows.

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 = 4 × 3	Object orientation
vector	Scale	12 = 4 × 3	Object scale
rect3d	Bounding box	48 = 8 × 6	Object maximum bounding box

The structure of the object index is as follows. Considering expandability, the LOD degree of the object can vary depending on the purpose of the application.

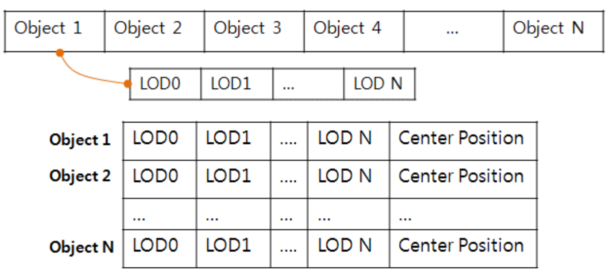


Figure 6. Object and LOD structure

Table 2 shows the structure of the head that manages general information on the LOD including the LOD degree and the start position and length of geometric information.

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

The information structure of the objects such as LOD is summarized in Table 3. General information on LOD and the start and end positions of the object vary according to the number of objects.

Table 3. Object information structure

Data Type	Name	Length (Byte)	Description
string	GUID (Global unique identifier)	22	Object GUID
long	Start position	4	Start position in file stream
long	End position	4	End position in file stream
LOD	LOD	Variables	LOD data stream

The structure of spatial indexing is not determined in this study; however, the area for relevant information is allocated in the format structure for further expandability.

The format structure of the geometry, summarized in Table 4, comprises a list of information that represents geometric shapes by a unique ID and LOD in the form of surface data comprising that of the peak, corner, and side.

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 Class Design

On the basis of the developed format, the class structure has been designed to be object-oriented as shown in Figure 7.

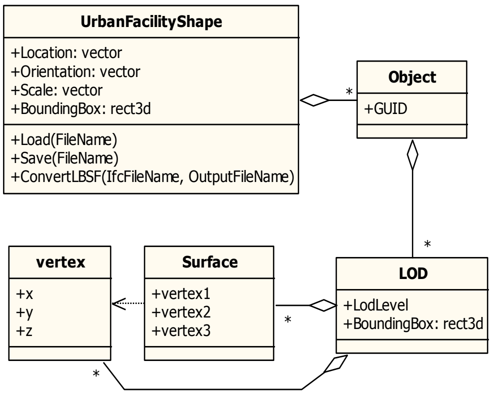


Figure 7. LBSF class structure (UML)

5.4 Design of the Component Structure

The component structure is demonstrated in Figure 8.

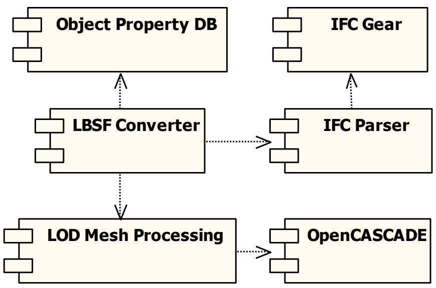


Figure 8. LBSF component structure

The main roles of each component are as follows:

Object Property DB: It separately manages the attribute data based on object and divides the data into an object GUID. This object GUID has the same value as that of the GUID of the BIM object and is used when it is connected with the properties of the geometry model in the LBSF.

IFC Parser: This component loads and parses the IFD file format, using the function of IFC Gear, which is an open source.

LBSF Converter: This component performs the mesh processing on the geometry of each parsed object by LOD and simplifies the mesh.

LOD Mesh Processing: This component performs mesh processing by LOD and partially uses the functions of OpenCASCADE, which are open source.

6 Comparison and Verification

With regard to the visualization performance, the proposed weight lightening method has been compared with the surface information conversion method, which has been proposed to handle information such as the file loading time, data volume, and memory usage; the method to extract the exterior surface; and the conventional method to use the IFC format.

Table 5. LOD general information structure

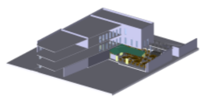
Data	Name	Shape Type
Sample #1	Cheongun University Main and Academic Information Center	
Sample #2	Family Restaurants	
Sample #3	Sungkyunkwan University dormitory building	

Table 6 lists the data characteristics of the samples used for performance comparison and verification. Although there are many ongoing BIM-based projects, it is practically difficult to use them as samples. Therefore, samples that have already been modeled and permitted for use in academic research have been selectively used in this study.

Table 6. Characteristics of Used Samples

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

The proposed methods are tested with samples with regard to the loading time, memory usage, and FPS indexes, which are considered important for visualizing large-volume BIM geometry. The time of loading IFC files, which is used as the performance measurement index, includes the time required for the parsing process and mesh processing of the geometry. However, the speed is expected to increase because these stages have already been processed in the lightweight BIM geometry format. The memory usage is influenced by a certain algorithm or by the use of the data format with regard to performance improvement. It is investigated to determine the influence of the proposed methods. FPS is related to the response rate; the lower the FPS value, the more difficult it is to observe the geometric information. In general, the higher the FPS value, the easier it is to observe the geometric information. Figure 9 shows a screenshot of the prototype program operation.

Figure 9. Prototype program screenshot

The results of the performance evaluation test conducted on the samples are summarized in Table 7.

Table 7. Test results of 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

According to the results of the quantitative performance evaluation, the data loading time of the LBSF format decreased by 95% in Sample #2. This implies that the performance was enhanced because the LBSF format performed parsing and the mesh processing in advance. The memory usage decreased by a maximum of 64% in Sample #1; the difference was insignificant in terms of the model data volume and FPS. This is because the amount of surface data required to be rendered for the two formats is constant. Therefore, the LBSF format that calculated the surface data in advance significantly improved the data loading time.

The LOD model data volume of the LBSF, which processed the exterior surface data at a level lower than that of LOD3, decreased by ~89% in Sample #1. The data loading time decreased by ~98% in Sample #2; the memory use decreased by ~88% in Sample #1; and FPS improved ~1.9 times in Sample #3. The general performance was enhanced significantly because the amount of exterior surface data to be rendered decreased.

The format performance of each sample has been analyzed in the form of graphs as shown in Figures 10 through 13. In general, when the BIM data of the IFC format are converted into the LBSF model, the loading time and memory usage rapidly decrease, whereas FPS improves. The result is consistent with the expectation prior to the experiment.

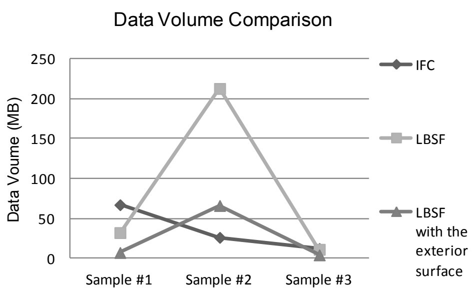


Figure 10. Samples data volume comparison of each format

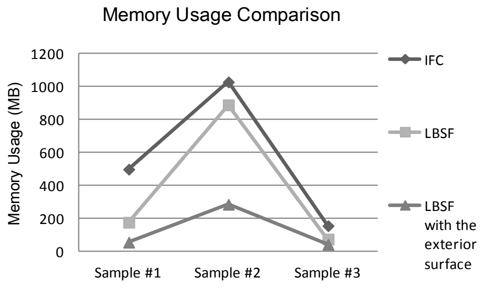


Figure 11. Samples data loading time comparison of each format

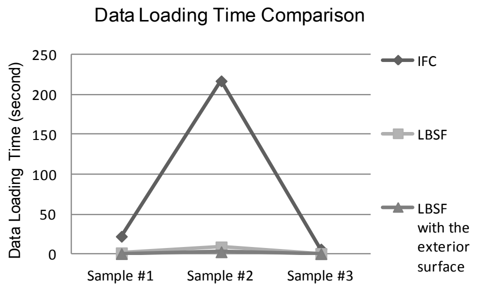


Figure 12. Samples memory usage comparison of each format

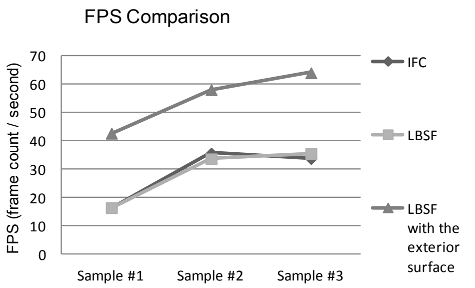


Figure 13. Samples FPS comparison of each format

7 Conclusion

This study proposed a method and a format structure that represents lightweight BIM geometry objects to solve performance-related problems encountered when the geometries of numerous BIM objects are visualized using GIS, which is necessary from the building facility management perspective. Accordingly, the prototype was realized, and the effectiveness of the geometric information visualization was confirmed on the basis of performance improvement using three samples and four indexes.

The comparison and verification results indicate that it is difficult to effectively represent a large number of BIM shapes that are expressed on a large area such as GIS, when the IFC format including all the information required for modeling is used from the building facility management perspective. In this case, it is beneficial to use the format that extracts the necessary shapes and properties from the perspective of facility management. Furthermore, the performance of the proposed LBSF format is better than that of the IFC format. In particular, the LBSF format that has only the exterior surface data is more effective for rendering all the BIM objects with regard to the GIS performance. Therefore, each format should be strategically used from the user and LOD perspective.

In our future work, mesh processing, in which the exterior surface data is handled, will be improved by considering the spatial index for visualization performance improvement. Moreover, the low number of samples, which is the limitation of this study, will be improved, and studies on automatically generating various LODs, which are required for visualization using GIS, will be conducted. Furthermore, the methods for data compression and virtual memory management will be studied to develop a distributed processing method for large-volume BIM objects using GIS.

Acknowledgements

This research was supported by a grant from the Strategic Research Project (Development of BIM/GIS Interoperability Open-Platform 2014) funded by the Korea Institute of Construction Technology.

References

[1] Chang K. J. A study on the expression of lightweight-visualization about CAE/CAI data and collaboration in a network environment, Master Thesis, Chung-Ang University, 2011.

[2] Hwang J. R., Kang T. W., and 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, 2012.

[3] Kwak J. G. Theoretical study on construction of 3D data for control logic validation, Doctoral Thesis, Ajou University, 2011.

[4] Kang T. W., Hong C. H., Hwang J. R., and Choi H. S. The external BIM reference model suggestion

for interoperability between BIM and GIS, Korea Spatial Information Society, 20(5):91–98, 2012.

[5] Lee O. L. Lightening CAE analysis data using JT, Master Thesis, Hanyang University, 2009.

[6] Park K. H. Automatic 3D PDF generation on a 3D viewer system, Master Thesis, Chung-Ang University, 2008.

[7] Reed K., Harrod D., and Conroy W. The Initial Graphics Exchanges Specification (IGES) Version 5.0. NIST, 1990.

[8] Thomas L. IFC4 - The new buildingSMART Standard, 2013.

[9] UGS, JT File Format Reference Version 8.1, 2006.

[10] Wakita A., Yajima M., Harada T., Toriya H., and Chiyokura H., XVL, A compact and qualified 3D representation, 2000.

[11] Gilles G. and Alexis M. 3D visualization of urban data based on CityGML with WebGL, *International Journal of 3-D Information Modeling*, 1(3):1–15, 2014.

[12] Chávez G. and Ávila F. Lightweight visualization for high-quality materials on WebGL, In *Proceedings of the 18th International Conference on 3D Web*, 109–116, 2013.