

건축물 평면 형상에 대한 형상-to-BIM 맵핑 규칙 정의

강태욱

한국건설기술연구원 미래융합연구본부

Geometry-to-BIM Mapping Rule Definition for Building Plane BIM object

Tae-Wook Kang

Department of Future Technology and Convergence Research, Korea Institute of Civil Engineering and Building Technology

요 약 최근 유지보수 등의 목적으로 다양한 건설 및 건축 분야에서 스캔 프로젝트가 수행되고 있다. 스캔된 결과로 생성된 포인트 클라우드는 스캔 대상물을 표현하는 수많은 점들로 구성되어 있다. 이런 스캔 데이터에서 치수를 포함한 필요한 정보를 추출하는 과정을 역설계라 한다. 포인트 클라우드를 BIM으로 모델링하는 역설계 과정은 수많은 수작업이 포함되어 있다. 시간이 많이 소모되는 역설계 작업 특성상 설계변경 같은 재작업 요청이 발생되면 비용은 기하급수적으로 증가한다. 역설계 자동화 기술이 이런 문제를 개선하는 데 도움을 줄 수 있다. 하지만, 역설계 산출물은 유스케이스에 따라 가변성을 가지며, 산출물의 종류와 상세수준은 달라질 수 있다. 이런 점을 고려해, 본 연구는 건축물 평면객체 기본 형상(primitive geometry)에서 BIM객체로 자동 맵핑하는 G2BM(Geometry-to-BIM mapping) 규칙 정의 방법을 제안한다. G2BM는 사용자 활용사례 가변성을 고려한 건축물 평면 BIM객체 역설계 프로세스 정의와 사용자화 방안을 제안한다. 그리고, 프로토타입을 통해 이에 대한 효과를 확인한다.

Abstract Recently, scanning projects have been carried out in various construction and construction fields for maintenance purposes. The point cloud generated by the scan results is composed of a number of points representing the object to be scanned. The process of extracting the necessary information, including dimensions, from such scan data is called paradox. The reverse engineering process of modeling a point cloud as BIM involves considerable manual work. Owing to the time-consuming reverse engineering nature of the work, the costs increase exponentially when rework requests are made, such as design changes. Reverse engineering automation technology can help improve these problems. On the other hand, the reverse design product is variable depending on the use, and the kind and detail level of the product may be different. This paper proposes the G2BM (Geometry-to-BIM mapping) rule definition method that automatically maps a BIM object from a primitive geometry to a BIM object. G2BM proposes a process definition and a customization method for reverse engineering BIM objects that consider the use case variability.

Keywords : 3D Point Cloud, Reverse Engineering, BIM, Geometry, Mapping, Rule, G2B-MD

This research was supported by a grant (19AUDP-B127891-03) from the Architecture & Urban Development Research Program funded by the Ministry of Land, Infrastructure and Transport of the Korean government.

*Corresponding Author : Tae-Wook Kang(Korea Institute of Civil Engineering and Building Technology)

email: laputa99999@gmail.com

Received May 8, 2019

Revised July 8, 2019

Accepted October 4, 2019

Published October 31, 2019

1. 서론

최근 유지보수 등의 목적으로 다양한 건설 및 건축 분야에서 스캔 프로젝트가 수행되고 있다[1].

LiDAR(Light Detection and Ranging)와 같은 스캐너로부터 스캔 데이터를 획득하는 과정을 스캐닝(scanning)이라 한다. 스캔 데이터는 포인트 클라우드(point cloud)라 하며, 속성 정보가 포함된 스캔 포인트(point)의 집합이다.

포인트 클라우드를 BIM(Building Information Modeling)으로 모델링하는 과정은 수많은 수작업이 포함되어 있다. 시간이 많이 소모되는 역설계 작업 특성상 설계변경 같은 재작업 요청이 발생되면 역설계 비용은 기하급수적으로 증가할 수 있다. 게다가, BIM 과정에서 실수나 착오로 얻은 데이터가 입력되면, 전체 작업 산출물이 무용지물이 되며, 다시 재작업해야 한다.

이 연구는 이런 수작업 기반 역설계를 자동화하여 효율을 개선할 목적으로 형상 역설계 기술의 일종으로 건축물 평면 기본 형상(primitive geometry)에서 BIM으로 맵핑하는 방법을 기술할 수 있는 G2BM 규칙 정의 방법을 제안한다.

본 연구는 건축물 평면 형상을 BIM 객체로 맵핑하는 방법에 국한한다. 건축물 평면객체는 벽체, 바닥, 천장, 문, 창호와 같이 평면이 기본 형상인 객체를 의미한다. G2BM은 사용자 유스케이스 관점을 고려해 평면 형상에서 BIM객체를 생성할 수 있다. G2B-MD(Mapping Definition)는 역설계 프로세스를 사용자화하여 다양한 응용 목적에 부합할 수 있는 가변성을 지원한다.

본 연구는 스캔 데이터인 3차원 포인트 클라우드 데이터(point cloud data, PCD)에서 평면 형상을 역설계하여 추출했음을 가정한다.

2. 연구 방법

형상에서 BIM 객체로 맵핑하기 위해, 먼저 관련 기술 동향을 분석하고 차별성을 확인한다. 이를 바탕으로 G2BM 시스템 구조와 핵심 컴포넌트를 정의하고, 규칙 기반으로 BIM 객체를 생성하기 위한 연산자를 정의한다. 마지막으로 제안한 G2BM 개념을 바탕으로 프로토타입을 구현하고 효과와 한계를 확인한다.

3. 관련연구

3차원 이미지 스캔은 수작업 중심의 작업 프로세스를 가지고 있다. 이 장에서는 역설계를 자동화하기 위한 연구를 살펴보고 이와 차이점을 살펴본다.

리모델링 프로젝트에 역설계 기술을 적용한 연구가 있었다. 이 연구는 레이저 스캐닝 정보 획득 방안을 도출하는 것에 목적이 있다[2].

역설계 기반 MEP 시설물 작업 관리 개선 방안 도출에 대한 연구가 있었다[3]. 이 연구는 역설계를 통한 생산성 향상 효과를 분석하는 데 초점을 맞추고 있다.

포인트 클라우드를 이용해 MEP 객체를 역설계하는 시스템에 대한 연구가 있었다[4]. 이 연구는 MEP 역설계를 위한 시스템 구조를 분석하는 것에 목적이 있다.

대용량 포인트 클라우드에서 파이프를 추출하는 데 필요한 알고리즘에 대한 연구가 있었다[5]. 이 연구는 MEP 객체 중 파이프 형상을 추출하는 알고리즘 개발에 중점을 두고 있다.

이미지 스캔을 기반으로 한 MEP 시설물 관리에 대한 연구가 있었다[6]. 이 연구는 배관과 같은 설비에 대한 역설계로 본 연구와는 차이가 있다.

이외에 역설계 및 스캔 기술 응용 방법에 대한 연구가 있었다[7, 8]. 역설계 시 세밀도 수준 정의에 대한 연구[7]는 역설계 지침에 대한 내용을 다루고 있으며, 스캔 기법[8]에 대한 연구는 드론, 고정식 스캔 등 작업 방법을 다루고 있어, 본 연구와는 차이가 있다.

앞에서 조사된 연구들은 크게 역설계 지침, 응용, MEP 역설계와 관련된 것으로 건축물 평면 형상을 BIM으로 맵핑하는 본 연구와는 차이가 있다. 본 연구는 비정형인 건축물 평면객체 기본 형상(primitive geometry)에서 BIM객체로 자동 맵핑하는 G2BM(Geometry-to-BIM mapping) 규칙 정의 방법을 제안한다.

4. G2BM 시스템 정의

4.1 시스템 구조

G2BM은 형상에서 형상의 특징을 추출하고, 이를 바탕으로 가장 유사도가 높은 BIM 라이브러리를 검색하여, 특징과 부합하는 형상에 대한 BIM 객체를 생성한다. 이를 위해, Fig. 1과 같은 G2BM(Geometry-to-BIM mapping) 시스템 컴포넌트가 필요하다.

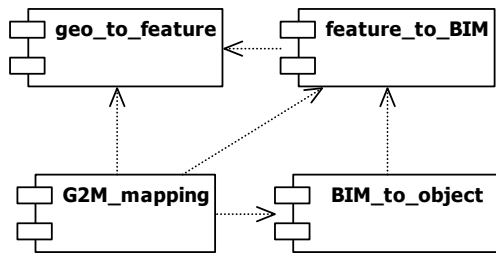


Fig. 1. G2BM system component

Table. 1은 G2BM 컴포넌트 역할을 정의한 것이다.

Table. 1. G2BM component role definition

Name	Role
G2M_mapping	<p>Performs geometry-to-BIM mapping. G2M_mapping provide a way to perform a mapping workflow. In the below mapping grammar, call specifies the mapping definition of each subcomponent that should be called.</p> <pre> G2M={call*, if, begin, end} *=multiple call={domain, name} domain={G2F-MD G2F F2B B2O} if={condition} condition={cp, {and or, cp}*} cp={left var, c-op, right var} c-op={'<', '>', '<=', '>=', '==', '!='} begin = call Indicates the beginning. end = end of call </pre>
geo_to_feature mapping definition (G2F-MD)	<p>To define the shape semantics from the point of view of use, we provide a method of defining feature extraction. define geometry to feature mapping, and provide the necessary operators.</p> <pre> G2F-MD={name, footprintPlane, LCS, planeShapeFeature, setGeometryFeature} footprintPlane = Floor Plane information for planar objects planeShapeFeature = Dimension information for planar objects setGeometryFeature = Set geometry feature information </pre>
geo_to_feature mapping (G2FM)	<p>Provides operators for mapping features of defined shapes to dimensions.</p>
feature_to_BIM (F2BM)	<p>Provides an operator that maps the features of the extracted shape to the dimension attributes.</p>
BIM_to_object (B2OM)	<p>Provides an operator to convert to a BIM object using geometry features mapped to dimension.</p> <pre> attributes.B2O={name, object*} object={type, parameter, constraint*} constraint={var*} var={name, value} </pre>

4.2 G2BM 알고리즘

형상에서 BIM객체를 추출하는 매핑 과정은 프로젝트 목적과 유스케이스(usecase)에 따라 다르다. G2BM 기술은 맵핑 함수를 정형화해서, 맵핑 과정을 사용자화할 수 있어야 한다.

Fig. 2는 G2BM 알고리즘 프로세스를 기술한 것이다. G2BM 알고리즘의 각 단계는 다음과 같다.

P1. Geometry-to-Feature mapping

PCD에서 맵핑된 형상은 단순한 polygon, polyline 형태이며, 형상의 의미를 추정할 수 있는 정보가 전혀 포함되어 있지 않다. 이런 이유로, 형상에서 의미를 추정할 수 있는 특징을 정의하는 방법이 필요하다.

평면 객체의 의미를 추정할 수 있도록 G2F-MD (geometry to feature mapping definition) 연산자를 제공한다. 이를 이용해, 역설계 사용 목적에 맞는 평면 객체 특징을 정의한다.

P2. Feature-to-BIM mapping(F2BM)

추출된 형상의 특징에서 BIM으로 맵핑 방법을 제공한다. 모든 형상은 현장 상태와 노이즈를 반영하므로, 이상적으로 디자인된 BIM객체 유형과 일치하지 않는 모호성 (ambiguous) 문제가 발생한다. 이는 어떤 형상을 어떤 BIM 객체와 1:1 매칭할 지 혼란스럽게 한다. F2BM은 규칙 기반 맵핑 문법을 제공해 모호성 문제를 해결한다. 모호성이 해결되면, 1:1로 형상을 BIM으로 맵핑할 수 있다.

P3. BIM-to-Object mapping(B2OM)

모호성 문제가 해결된 BIM을 특정 모델러에 객체로 생성할 수 있는 방법을 제공한다. 모델러는 BIM 모델에 관한 여러 가지 정보를 포함하고 있으며, 이는 그 모델러 시스템 외부에서 알기 어려운 경우가 많다. 예를 들어, 벽체를 제약하는 층고(storey height)는 전체 BIM모델을 알지 못하면 해결하기 어려운 문제이다. B2OM은 실제 벽체를 모델러 내에서 생성할 때 필요한 정보를 제공하는 방법을 정의한다.

B2GM은 제시된 맵핑 연산자를 이용해 역설계 맵핑 프로세스를 정의하므로, 다음과 같이 XML이나 python과 같은 언어로 구현될 수 있다.

```
<G2M name='plane-to-wall'>
```

```
<call domain='G2F-MD' name='plane_semantics'/>
```

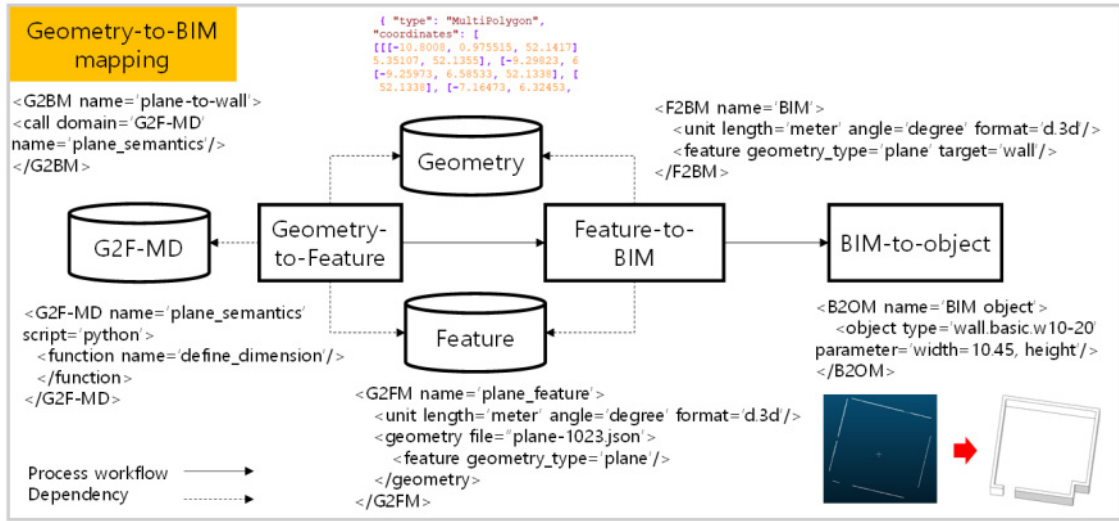


Fig. 2. G2BM algorithm process

```

<call domain='G2FM' name='plane_feature' />
<call domain='F2BM' name='BIM' />
<call domain='B2OM' name='BIM_object' />
</G2M>
  
```

```

</function>
</G2F-MD>
  
```

형상의 특징을 치수로 맵핑하기 위한 연산자는 다음과 같이 정의될 수 있다.

```

G2F={name, unit, geometry*}
unit={length, angle}
length=length unit
angle=angle unit
geometry={feature*}
feature={geometry type}
dims={dim*}
dim={name, value, type}
  
```

5. G2FM 연산자

5.1 형상 특징 추출

이 장에서는 G2M을 위해 형상에서 특징을 추출하는 G2FM(Geometry-to-BIM Mapping) 규칙 정의 방법을 기술한다. 맵핑 조건으로 사용해야 할 데이터가 서로 1:1이 아닌 이유로 형상에서 BIM으로 직접적인 맵핑은 어렵다. 이런 이유로, 형상에서 추출한 특징과 가장 유사한 BIM 객체로 맵핑하는 절차를 규칙 기반으로 정의한다.

이를 위해, Table. 2와 같이 형상에서 맵핑할 특징 데이터를 추출하는 G2F-MD 연산자를 정의한다.

다음은 G2F-MD operator를 이용해 평면 형상에서 치수값을 추출하는 방법을 정의한 예이다.

```

<G2F-MD name='plane_semantics' script='python'>
  <function name='define_dimension'>
    footprintPlane = footprintPlane(geometry)
    LCS = footprintPlane.LCS()
    dims = planeShapeFeature(geometry, LCS)
    for d in dims:
      setGeometryFeature(geometry, d.name, d.value)
  </function>
</G2F-MD>
  
```

이 연산자는 XML, python과 같은 다양한 실행 가능한 형식으로 표현될 수 있다.

```

<G2FM name='plane_feature'>
  <unit length='meter' angle='degree' format='d.3d' />
  <geometry file='plane-1023.json' />
  <feature geometry_type='plane' />
  <dim name='width' type='real' value='10.254' />
  <dim name='height' type='real' value='3.532' />
  <dim name='PCA' type='complex' value='((0,0,1), (1,0,0), (0,1,0))' />
</feature>
</geometry>
</G2FM>
  
```

Table. 2. G2F-MD operator

Operator	Input	Output
dims (dimension s)	{index name} index=number name=string	{dims} dims={dim*} dim={var_name, var_value, [var_type]} *=multiple []=option
footprintPlane	{geometry, LCS} LCS=local coordination system	{footprintPlane} footprintPlane={planeCoefficient , origin, heightVector} planeCoefficient={A, B, C, D} origin={x, y, z} heightVecgtor={v1, v2, v3}
planeShape Feature	{geometry, footprintPlane} constraint(geometry.type == polygon)	dims={width, height}
geometryPCAFeature	{geometry}	dims={basicVector} basicVector={v1, v2, v3}
setGeometryFeature	{geometry, name, value} name=dim name value=dim value	none

5.2 BIM 맵핑

이 장에서는 추출된 형상의 특징을 치수 속성으로 맵핑하는 F2B 연산자를 정의한다. 이 연산자에서 feature는 허용오차, 맵핑 조건 및 조건에 따라 맵핑되는 BIM 라이브러리 type을 지정한다.

```
F2B={name, unit, feature*}
feature={tolerance, if}
tolerance={length, angle}
if={condition, BIM}
condition={cp, {and | or, cp}*}
cp={left var, c-op, right var}
cp-op={'<', '>', '<=', '>=', '==', '!='}
BIM={type, parameter}
```

이 연산자는 XML, python과 같은 다양한 실행 가능한 형식으로 표현될 수 있다.

```
<F2BM name='BIM'>
<unit length='meter' angle='degree' format='d.3d'/>
<feature geometry_type='plane' target='wall'>
<tolerance length=0.05 angle=0.1/>
<if condition='source.width > 10.0 and source.width
```

```
<= 20.0'>
```

```
<BIM type='basic.w10-20' parameter='width=
mod(width, 0.05), height'/>
```

```
</if>
```

```
<if condition='source.width <= 10.0'>
```

```
<BIM type='basic.w10' var='width=10.0, height'/>
```

```
</if>
```

```
</feature>
```

```
</F2BM>
```

치수 속성으로 맵핑된 형상 특징은 다음과 같은 B2O 연산자를 통해 BIM object로 변환된다.

```
B2O={name, object*}
object={type, parameter, constraint*}
constraint={var*}
var={name, value}
```

이 연산자는 다음과 같이 XML, python과 같은 다양한 실행 가능한 형식으로 표현될 수 있다.

```
<B2OM name='BIM_object'>
<object type='wall.basic.w10-20' parameter='
width=10.45, height'>
<constraint>
<var name='height' value='storey.height'/>
</constraint>
</object>
</B2OM>
```

6. 프로토타입 개발

이 장에서는 앞서 기술된 G2BM 알고리즘을 일부 구현해 보고, 그 효과를 확인해 본다. 데이터는 Fig. 3과 같이 도심재생과 관련해 S상가에서 스캐닝을 수행하였으며, 이를 정합한 포인트 클라우드를 사용하였다. 이를 형상 모델링한 결과를 이용해 BIM 으로 맵핑하는 과정을 수행하였다. 맵핑에 활용되는 G2BM 스크립트는 파이썬으로 구현하였다.

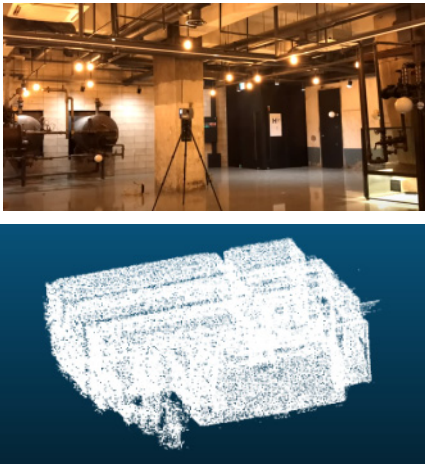


Fig. 3. S-commercial building scanning and results

포인트 클라우드 뷰어는 CloudCompare를 사용하였으며, G2BM으로 변환된 BIM객체를 생성하는 도구는 Revit을 사용해 Addin으로 개발되었다.

Fig. 4는 형상으로부터 치수 등 특징을 추출하기 위해 포인트 클라우드로부터 자동으로 세그먼테이션할 결과이다. 벽체와 바닥이 추출된 것을 확인할 수 있다. 추출된 벽체와 바닥은 polyline 형식으로 변환되고, 여기서 치수 등을 형상 특징을 추출할 수 있다.

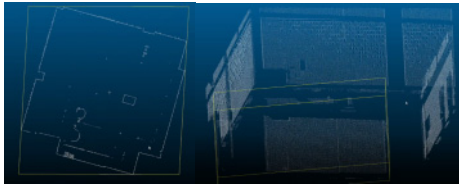


Fig. 4. Geometry segmentation from 3D point cloud

Fig. 5는 형상에서 추출된 특징을 이용해 Revit에서 벽체, 바닥, 천정 객체로 자동 맵핑한 결과이다.

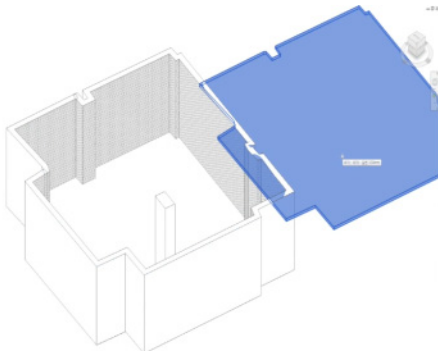


Fig. 5. BIM model from geometry

실제 BIM 평면 객체까지 얻는 데 투입된 노력은 수작업에 비해 효과적이었다. 다만, 맵핑 객체의 모서리 등 표시 정확도면에서 이슈가 있어 개선이 필요함을 알 수 있었다. 다만, 이런 문제는 데이터에서 형상을 변환할 때 문제이다.

7. 결론

본 연구는 스캔 형상을 BIM 객체로 맵핑하기 위해, 관련 기술 동향을 분석하고 차별성을 확인하였다. 이를 바탕으로 G2BM 시스템 구조와 핵심 컴포넌트, 규칙 기반으로 BIM 객체를 생성하기 위한 연산자를 정의하였다. 마지막으로 제안한 B2GM 개념을 바탕으로 프로토타입을 구현하고 효과와 한계를 확인하였다.

프로토타입을 통해 S상업건물을 스캐닝하고 형상으로 변환된 데이터셋을 사용해 테스트해보았다. 이를 통해, BIM 평면 객체까지 얻는 데 투입된 노력은 수작업에 비해 효과적임을 확인할 수 있었다.

향후, 스캔 테스트 사례를 더 추가해 알고리즘 효과를 정량적으로 분석한 후, 맵핑 정확도를 좀 더 개선하고, 맵핑 대상이 되는 평면 형상의 종류를 좀 더 다양화할 계획이다.

References

- [1] T.W. Kang, "BIM-based Smart Facility Management Framework for Existing Buildings", *Review of Architecture and Building Science*, Vol.62, No.6 pp.37-42, 2018.
- [2] S.S. Lee, S.W. Kwon, "A Study on Optimal Laser Scanning method for Reverse Engineering at Interior Remodeling Project", *Korea Journal of Construction Engineering and Management*, Vol.15, No.3, pp.3-11, 2014.
DOI: <http://dx.doi.org/10.6106/KJCEM.2014.15.3.003>
- [3] T.W. Kang, J.E. Kim, T.S. Jung, "Study on 3D Reverse Engineering-based MEP Facility Management Improvement Method", *Journal of the Korea Academia-Industrial Society*, Vol.17, No.8, pp.38-45, 2016.
DOI: <http://dx.doi.org/10.5762/KAIS.2016.17.8.38>
- [4] T.W. Kang, "System Architecture for Point Cloud-based Reverse Engineering of Architectural MEP Object", *Korea Journal of Construction Engineering and Management*, Vol.2, pp.1-10, 2014.
DOI: <http://dx.doi.org/10.5762/KAIS.2014.15.9.5870>

- [5] J.E. Kim, "Large Point Cloud-based Pipe Shape Reverse Engineering Automation Method", *Journal of the Korea Academia-Industrial Society*, Vol.17, No.3, pp.692-698, 2016.
DOI: <http://dx.doi.org/10.5762/KAIS.2016.17.3.692>
- [6] T.W. Kang, "Study on 3D Image Scan-based MEP Facility Management Technology", *Korea Institute of BIM*, Vol.6, No.4, pp.18-26, 2016
DOI: <http://dx.doi.org/10.13161/kibim.2016.6.4.018>
- [7] J.H Chae, J.Y. Lee, "Definition of 3D Modeling Level of Detail in BIM Regeneration Through Reverse Engineering - Case Study on 3D Modeling Using Terrestrial LiDAR", *Korea Institute of BIM*, Vol.7, No.4, pp.8-20, 2018
DOI: <https://doi.org/10.13161/kibim.2017.7.4.008>
- [8] K.R. Jung, B.D. Lee, T.H. Kim, "Engineering Construction With High-Precision Survey Equipment (3D laser scanner, 3D photo scanner, drone)", *The Korea Institute of Building Construction*, Vol.9, No.1, pp.133-134

강 태 욱(Tae-Wook Kang)

[정회원]



- 2009년 3월 : 중앙대학교 건설환경공학 (공학박사)
- 2010년 6월 ~ 2011년 5월 : 중앙대 겸임교수
- 2011년 6월 ~ 2012년 6월 : 한길아이티 BIM본부장
- 2012년 7월 ~ 현재 : 한국건설기술연구원 연구위원

〈관심분야〉

CAD, CAM, BIM, GIS, Computer Graphics, SW공학