

# ZERO WATERMARKING SCHEME FOR CITYGML

<sup>1</sup>DAYOU JIANG, <sup>2</sup>JONGWEON KIM

<sup>1</sup>Department of Copyright Protection, Sangmyung University, Seoul, Korea

<sup>2</sup>Department of Electronics Engineering, Sangmyung University, Seoul, Korea

E-mail: <sup>1</sup>dyjiang@cclabs.kr, <sup>2</sup>jwkim@smu.ac.kr

## ABSTRACT

City Geography Markup Language (CityGML) is developed as an open data model, which is based on the XML format for the storage and exchange of virtual 3D city models. CityGML defines the basic entities, attributes, and relations of a 3D city model. With respect to the cost-effective sustainable maintenance of 3D city models, it allows the reuse of the same data in different application fields. However, the copyright protection in suchlike virtual 3D city files is not popular. Therefore, we present a zero watermarking scheme based on the global geometry property of the point cloud for CityGML. Firstly, the paper analyzes the data structure of CityGML and discusses the feasibility data for watermark. Secondly, extracts the point and topology information from CityGML. Finally, generates the zero-watermarking by using the global geometry property from vertex norm. We conduct several experiments on CityGML models with translation, rotation, uniform scaling, vertex re-ordering, simplification, smoothing, and noise attacks. Experiments show the scheme can satisfy the requests for copyright protection of CityGML models.

**Keywords:** *CityGML, Zero Watermarking, Point Cloud, Geometry Property*

## 1. INTRODUCTION

A 3D city model is a manifestation of an urban area with a 3D geometry of common urban objects and structures, in which buildings are the most prominent feature [1]. Nowadays, increasingly virtual 3D city models are being built for different applications, such as urban planning, disaster management, 3D cadaster, tourism, facility management, architectural models and drawings, procedural modelling, and environmental simulations [2]. Although each 3D application requires its own specific 3D data, the CityGML model can help link the requirements to specific applications. It is especially important that the CityGML is not only designed for the model visualization, unlike the vast majority of 3D city models that have been defined as purely graphical or geometrical models, it also considers the semantic and topological aspects, which can be used for thematic queries, analysis, and spatial data mining.

Due to the growing demand for applications such as city planning, and architectural design, the construction and protection of virtual 3D city models are quick-off required from governments and companies. So the profit brought by CityGML is increasing at very fast speed. Owing

to their reusability and transportability characteristics, CityGML is widely used and easily modified for commercial profit without the author's permission. Thus, the development of a copyright protection technology for CityGML is an urgent problem to be solved, which is significant for the wholesome virtual 3D city models market environment.

Over the last decade, a large number of articles on CityGML have been published. However, few researches on CityGML copyright protection technology have been reported so far. Most of those articles are focused on its underlying concepts, its applications, and its relationship to other standards. Gröger [3], the editor of CityGML, gives an overview of CityGML, its underlying concepts, its Levels-of-Detail, how to extend it, its applications, its likely future development, and the role it plays in scientific research. Fan [4] presents an approach for deriving LoD2 buildings from LoD3 models which are essentially the exterior shells of buildings without opening objects. The approach treats different semantic components of a building separately with the aim to preserve the characteristics of ground plan. Marcus [5] explained a method for the creation of highly detailed CityGML LoD4 models with interior structures. The paper further continues on the

automated generation of CityGML models from OpenStreetMap. Baig [6] proposed a unified generalization framework to derive multiple LoDs (LoD3-LoD1) taking both semantics and geometric aspects of 3D buildings modelled in CityGML. Biljecki [7] provided a list of identified use cases of 3D city models and their applications. The study demonstrates that 3D city models are employed in at least 29 use cases that are a part of more than 100 applications. Jusuf [8] used Autodesk Revit and Graphisoft Archicad in developing the building models and used Feature Manipulation Engine (FME) to restructure the data model (IFC) and transform it to the destination data format (CityGML).

Since the components of CityGML datasets are similar to 2D vector and 3D mesh. Consequently, the inspiration of copyright protection technology on CityGML can be drawn from those two related digital copyright technologies. For 2D vector watermarking, Abubahia [9] addressed the simplification and interpolation modifications by exploiting the bounding box of vector data. Lin [10] used spider web by adding vertices for entities to ensure the robustness from watermark embedding in the stream order. The method is a semi-blind and semi-fragile reversible watermarking scheme. For 3D mesh watermarking, Liu [11] used multiresolution adaptive parameterization surface (MAPS) to categorize the vertices of the 3D model into two levels to instead of using the original model during the watermark extraction process. The coarse level vertices are used to establish an invariant space and some fine level vertices are selected to embed the watermark. Zafeiriou [12] embedded the watermark in a set of vertices that correspond to a range of angles in

spherical coordinate system's  $\Omega$  domain. Baskurt [13] used the analytic and continuous geometric volume moment to cover the watermark. During watermark embedding, first using its global volume moments to normalize the input mesh to a canonical and robust spatial pose; then, decompose the normalized mesh into patches; finally embed the watermark through a modified scalar Costa quantization. Wang [14] used multi-features like shape diameter function skewness vector, vertex normal skewness vector, and ratio vector. Combined with the distribution feature of the number of stable vertices, a resilient watermark is applied.

It is necessary for researchers to explore the blank fields of copyright protection on CityGML models. This paper focuses on the data structure of CityGML, and analyses the feasibility of the watermarking technique, finally proposes a zero-watermarking scheme for CityGML models.

The remainder of this paper is organized as follows: Section 2 briefly reviews the related works; Section 3 presents the watermark embedding and extraction procedures; Section 4 presents some experimental results; Section 5 concludes the paper and proposes several future working directions.

## 2. RELATED WORK

### 2.1 Data Structure of CityGML

CityGML datasets consist of a set of XML files and possibly some accompanying images as textures. Fig.1 shows the visualized CityGML model. In the left image, each rectangular bounding box represents an object in the model. From the right image, we can know that the polygons of each object have a big difference.

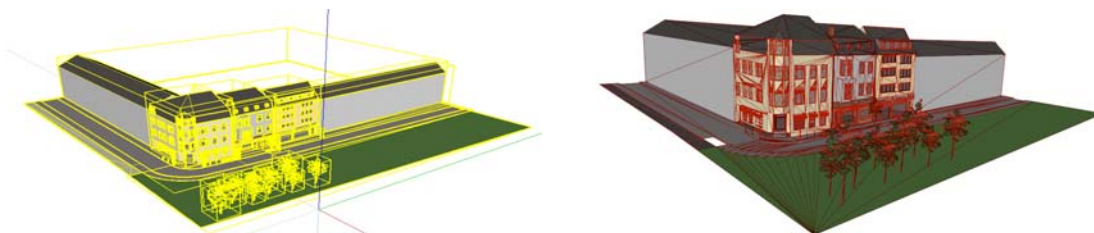


Figure 1: Visualized CityGML Model (left with boxes, right with polygons)

Assembling the CityGML into thematic groups, it can be decomposed into a core module and thematic extension modules. All modules including the core are specified by their own XML schema file, each defining a globally unique XML namespace. The CityGML core

module clearly states the basic concepts and standard components of the CityGML model. It as the general lower bound of the entire CityGML data model and a dependency of all thematic extension modules is very necessary for design. The extension modules are based on the

core module, which contains a logically separate thematic component of the CityGML data model. The CityGML standard 2.0v introduced thirteen thematic extension modules, including Appearance, Bridge, Building, CityFurniture, CityObjectGroup, Generics, LandUse, Relief, Transportation, Tunnel, Vegetation, WaterBody and TexturedSurface.

CityGML mainly has two general characteristics:

1- The CityGML data model consists of class definitions for the most important types of objects within virtual 3D city models. Fig. 2 shows some core modules of CityGML model, such as Transportation, Tunnel, Bridge, WaterBody, Building, Vegetation, LandUse, and Cityfurniture.

2- CityGML distinguishes five consecutive Levels of Detail (LOD), where the geometry of the objects and the topic distinctions become more detailed as the LOD increases. The roughest level LOD0 is essentially a two and a half dimensional Digital Terrain Model, and usually an aerial image or a map may be draped over it. In Buildings, the LOD0 is represented by footprint or roof edge polygons. As shown in Fig. 3, in city building models, LOD1 shows buildings with flat roof structures, LOD2 has differentiated roof structures and thematically differentiated boundary surfaces, LOD3 denotes detailed wall and roof structures potentially including doors and windows, LOD4 adds the interior structures.

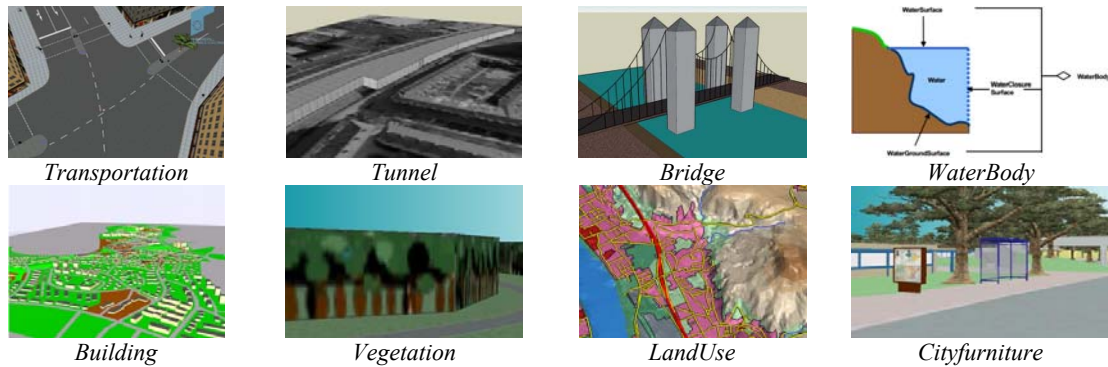


Figure 2: Samples Of The CityGML Core Modules

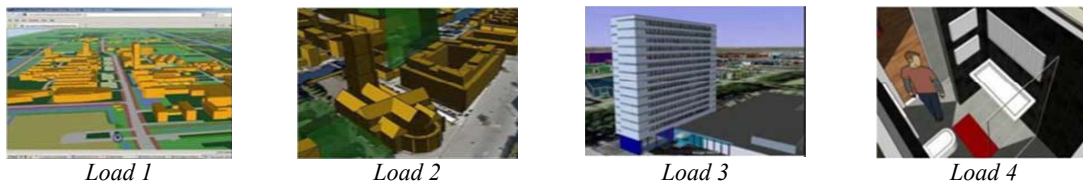


Figure 3: Examples For City Building Models In LOD1 ~LOD4

## 2.2 Feasibility of Watermarking for CityGML

Through analyzing the data structure of CityGML, in principle, the watermark can be embedded into the XML file and the textures images. Textures in CityGML are always raster-based 2D textures. The specification of texture wrapping is adopted from the COLLADA standard. Texture wrapping is required when accessing a texture outside the underlying image raster [2]. As shown in Fig. 4, most of those textures images' texture structure is simplex and the difference of pixels' brightness is very small. What's more, due to the WrapMode has five wrap modes (none, wrap, mirror, clamp, and

border) and the mapping function is complex, so the location and size of texture maps are not fixed, thus, embedding the watermark in the texture images is pretty a tough job.

The CityGML Schema uses the W3C XML schema language to describe the grammar of conformant CityGML data instances. It provides Schematron rules precisely describing referential integrity constraints on CityGML property elements denoting the relation between CityGML objects. Fig. 5 shows the sample of XML schema of CityGML model. The XML as a markup language, naturally, can be watermarked likes other markup languages based on the content or logical structure. Pradnya [15]

presented an XML watermarking based on the SHA-512 algorithm and data partitioning. Romaric [16] fixed the schema of the document by inserting constraints expressed in schema Tron and then by saving the structure of the XML tree using nodes cardinalities. The method can preserve the structure of the document and detect any modification that could occur to the file structure. Unlike multimedia data, XML data are diverse in nature: some are data-centric and numeric while some are document-centric and verbose. A major challenge in XML watermarking is the extraction of watermarked elements after modifications. Once the standard structure of XML file is changed for

watermarking, it will be hard for parsing XML. Moreover, the invisibility and robustness of the XML watermarking is awful.

Through parsing the XML file, the information of mass points, lines, faces, and polygons can be obtained. As shown in Fig. 5, the geometric information of model objects is defined in GML schema, and the boundary position of LowerCorner and UpperCorner of object are given. Considering the structure of the CityGML model, embedding the watermark in model's structural information has more meaningful than in XML files. Moreover, the robustness can be ensured.

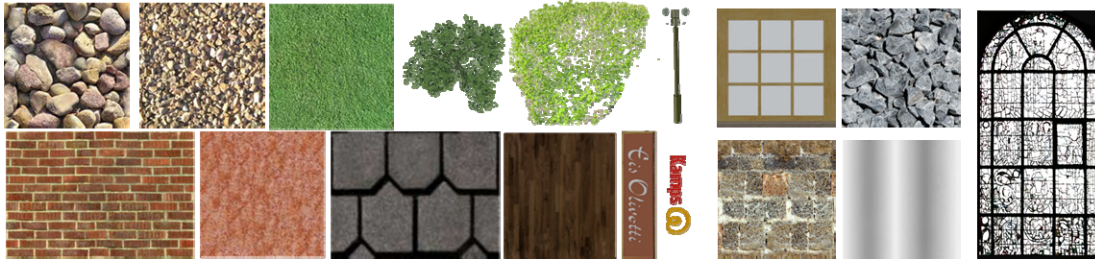


Figure 4: Samples Of CityGML Textures

```

<?xml version="1.0" encoding="UTF-8" ?>
<!--CityGML Dataset produced with CityGML Export Plugin for Sketchup by GeoRES -->
<!--http://www.geores.de -->
<core:CityModel xmlns="http://www.opengis.net/citygml/profiles/base/1.0" xmlns:core="http://www.opengis.net/citygml/1.0" xmlns:blq="http://www.opengis.net/citygml/1.0" xmlns:app="http://www.opengis.net/citygml/appearance/1.0" xmlns:gml="http://www.opengis.net/gml" xmlns:xAL="urn:oasis:names:tc:ciq:xdschema:xAL:2.0" xmlns:wtr="http://www.opengis.net/citygml/waterbody/1.0" xmlns:dems="http://www.opengis.net/citygml/relief/1.0" xmlns:luse="http://www.opengis.net/citygml/landuse/1.0" xmlns:veg="http://www.opengis.net/citygml/vegetation/1.0" xmlns:traa="http://www.opengis.net/citygml/transportation/1.0" xmlns:gen="http://www.opengis.net/citygml/generics/1.0" xmlns:schemaLocation="http://www.opengis.net/citygml/1.0 http://schemas.opengis.net/citygml/1.0/citygmlBase.xsd http://www.opengis.net/citygml/appearance http://www.opengis.net/citygml/cityfurniture/1.0 http://schemas.opengis.net/citygml/cityfurniture/1.0/cityfurniture.xsd http://www.opengis.net/citygml/transportation http://www.opengis.net/citygml/transportation/1.0 http://schemas.opengis.net/citygml/transportation/1.0/transportation.xsd http://www.opengis.net/citygml" >
  <gml:boundedBy>
    <gml:Envelope srsDimension="3" srsName="urn:ogc:def:crs:EPSG:31467">
      <gml:lowerCorner>3499943.625415 5399989.574130 0.0</gml:lowerCorner>
      <gml:upperCorner>3500026.358571 5400069.880516 20.0</gml:upperCorner>
    </gml:Envelope>
  </gml:boundedBy>
  <app:AppearanceMember>

```

Figure 5: Sample XML Code Of CityGML

### 2.3 Information Extraction of CityGML

The information of model's structures is stored in the XML files. So the CityGML dataset should be parsed firstly. XML files can be parsed by useful software CityGML4j [17] and converted into Java objects. Those Java objects contain both geometry and semantic information of City objects. The 3D City Database, Importer/Exporter, Web Feature Service, 3DCityDB-Web-Map-Client and citygml4j are usually combined used to represent, store and manage virtual 3D City models. As shown in

Fig. 6, the CityGML schema is bind to the CityGML4j object model. CityGML instance documents are deserialized into a tree of corresponding Java objects representing the content and organization of the instance document. After processing the CityGML data, the Java content tree can be easily serialized back into a CityGML instance document. To embed the watermark into the CityGML, we also need to analyze the the objects' geometric information of CityGML model.



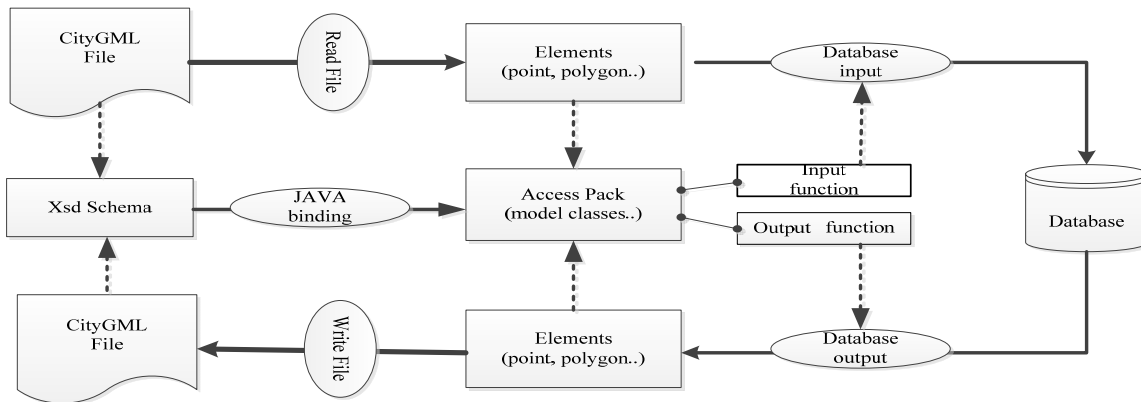


Figure 6: Input data And Output data Of CityGML

Fig.7 shows the geometry and geography data types of spatial data. The CityGML dataset also denotes those data types. Here, we select the point, normal and polygon information to study the watermark construction method. Fig. 8 shows the CityGML's texture image and its corresponding visualized 3D City object, the scatter plot of its points, and the 3D plot of its polygons. Ordinarily, the vertices in normal 3D models are nearly evenly distributed and its faces are all triangles. The area differences of those faces are not big. However, as seen in Fig. 8, in

CityGML model, the number of vertices is much smaller, the distance difference between adjacent vertices is relatively larger and the polygons are not the same type. From this, compared to the faces, the vertices can carry much more watermark information. Embedding the watermark by perturbing the coordinate values is relatively simpler than using faces and lines information. Due to the irregular faces of CityGML models, the normal, curvatures, and areas also hard to be used.

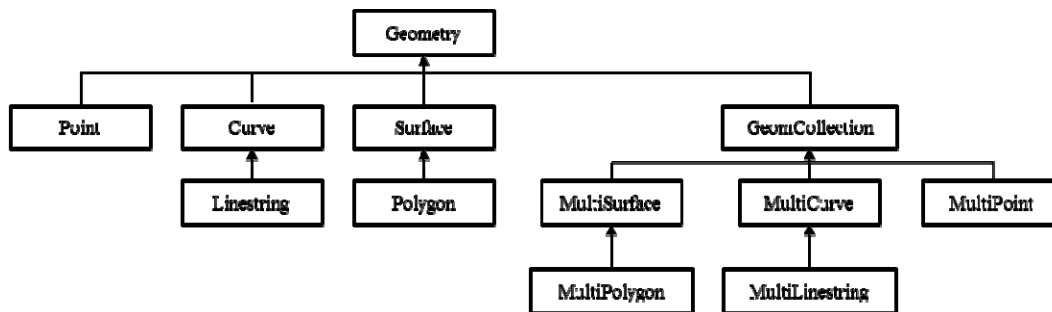


Figure 7: Geometry and Geography Data Types Of Spatial data

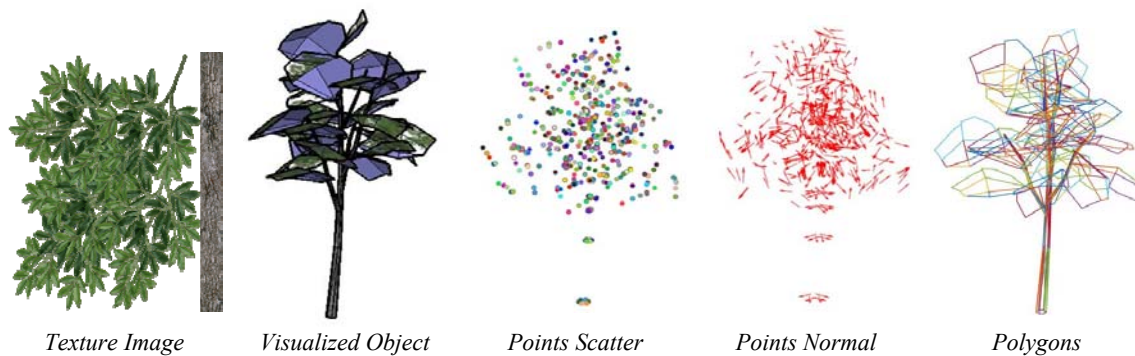


Figure 7: Sample Of CityGML Object And Its Information

### 3. PROPOSED ALGORITHM

#### 3.1 Zero Watermarking Construction

As has been analyzed above, here, we use vertices information for watermark construction. The vertex norms are calculated and ordered, the watermark is constructed by analyzing the distribution of the number of vertices in each partitions.

The vertex set  $V$  is defined as:

$$V = \{vi \in R^3 \mid 0 \leq i \leq Nv-1\} \quad (1)$$

Here,  $Nv$  is the number of vertices,  $vi = (xi, yi, zi)$  is the coordinates of a vertex. Define  $vc = (xc, yc, zc)$  as the model center, and the number of watermark bits to be constructed is  $Nw$ . To calculate and use the vertex norm of the vertex more easier, the Cartesian coordinates  $(x, y, z)$  of the vertices of the 3D model are transformed into spherical coordinates  $(\rho i, \theta, \phi)$ . Here, the spherical coordinate component  $\rho i$  is the vertex norms. The operation procedure of constructing zero-watermark is shown in Fig. 9. The watermark construction process is implemented according to the following steps:

1- Calculate the model center and transform the Cartesian coordinates to spherical coordinates.

2- Build vertex set according to the order of the vertex norm from small to large.

3- Normalize the vertex norm into the interval of  $[0, 1]$ .

4- Divide the vertices into  $Nw$  distinct partitions and calculate the numbers of vertices in each partion  $NVi$ .

5- Calculate the mean number of vertex  $MVa$  in the whole vertices set and construct the temporary watermark  $wt$  by comparing  $NVi$  with  $MVa$ .

6- Generate the temporary watermark  $wt$ .

Compute the model center by the formula:

$$vc = \frac{1}{N} \sum_{i=0}^{N-1} vi \quad (2)$$

Calculate the vertex norm by the formula:

$$\rho i = \sqrt{(xi - xc)^2 + (yi - yc)^2 + (zi - zc)^2} \quad (3)$$

Define the  $i$ -th vertex norm partition by the formula:

$$P_{norm,i} = \{\rho ij \mid \frac{1}{Nw} \cdot i < \rho k < \frac{1}{Nw} \cdot (i+1)\} \quad (4)$$

Here,  $\rho k$  is the  $k$ -th vertex norm. Generate the temporary watermark by the if-else conditional statement:

$$\begin{cases} NVi < MVa \{wt_i = 0\} \\ NVi \geq MVa \{wt_i = 1\} \end{cases} \quad (5)$$

7- Generate the final zero-watermarking  $wm$  by scrambling the temporary watermark with a key.

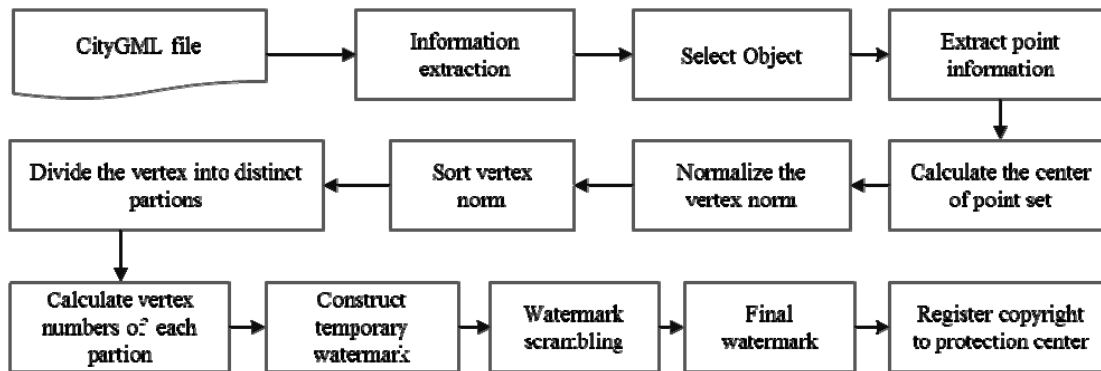


Figure 9: Flowchart Of Constructing CityGML Watermarking

#### 3.2 Zero Watermarking Detection

The zero-watermarking detection procedure involves watermark extraction and an examination of the similarity. The watermark extraction process is similar to the method of watermark construction. The watermarking detection is shown in Fig. 10.

Ordinarly, under the attacks such as nosing, rotation, translation, scaling, and vertex

reordering, only using the test model can detect whether the model has been copyright protected. However, under the attacks such as cropping and smoothing, the model needs resampling and relocated. Consequently, to enhance the robustness of the watermarking method, the resampling and relocated processing also conducted. It aims to obtain the similar partitions of test model in the watermark detection process

as partitions that have been divided in the original model in the watermark embedding process. In the watermark detection process, the original model can be used for resampling the test model. The extraction watermark to be examined is defined as  $w_d$ . The bit error rate (BER) between the extracted watermark and the

original watermark is calculated then the similarity value is compared with the preset BER threshold. If its value is less than the threshold, the examined CityGML model can be assumed has the registered original watermark and the copyright is verified, otherwise, the model is without permission.

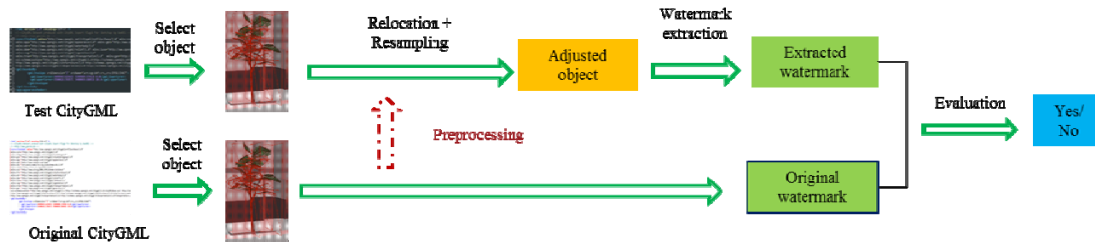


Figure 10: Flowchart Of Extracting CityGML Watermarking

#### 4. EXPERIMENTAL RESULTS AND ANALYSIS

We implemented the experiments by using Matlab2016a. The tested CityGML model `geoRES_testdata_v1.0.0.xml` is accessible from CityGML website [18]. We use an object in the model for the test. The object tree (GMLID\_SO1124524\_1571\_4314) has 535 vertices and 378 polygons. The length of the watermark is 64, and the threshold for watermark detection is 0.4 according to the experimental value.

To test the resilience of the proposed method, we carried on following attacks such as translation, rotation, scaling, noise addition, simplification, smoothing, and vertex reordering.

The original watermark and attacked models are shown in Fig. 11. In the paper, we adopted the BER metric to evaluate the difference of the original watermark and extracted watermark. The experiment results are listed in Table 1.

Table 1: Experimental Results of the experimental attacks

Attacks	BER (%)
Translation	0
Rotation 30°	0
Uniform Scaling	0
Noise 10e-9	0
Noise 10e-8	7.81
Vertex reordering	0
Simplification 30%	6.25
Smooth 5	4.69

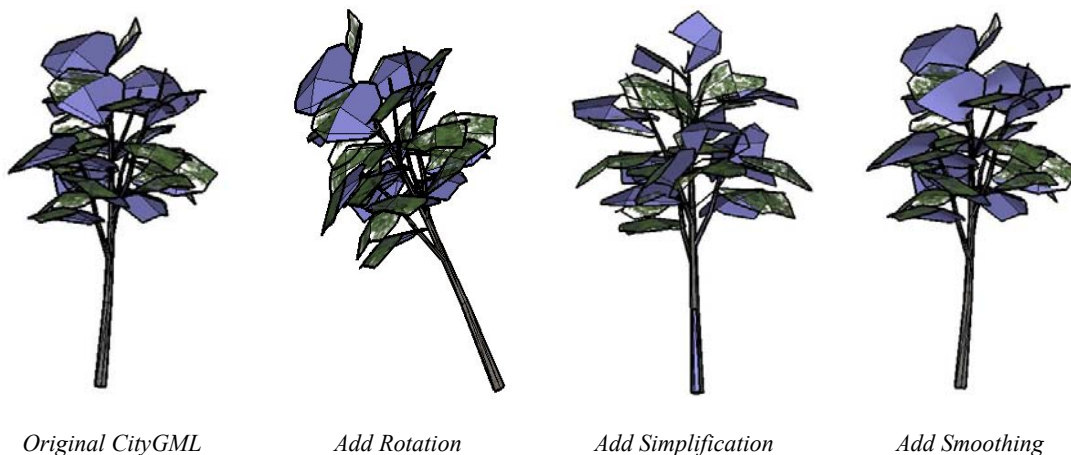


Figure 11: Original CityGML Model And Attacked Model

Because the global geometry property of vertex norm is calculated from the distance

between the vertices to the model center, therefore, it is not affected by either translation

or rotation; additionally the vertex norm was normalized before constructing the watermark, so that it will not be affected by the uniform scaling. With regard to vertex reordering, vertex norm is independent of the vertex order. For noise addition, the addition noises are randomly added to vertexes of the 3D model, because the coordinated values of CityGML have a big difference in different directions (as shown in Fig. 12), consequently the noising parameter

must be very small. The details of the model surface have been destroyed obvious but the watermark has no change. For simplification, the number of vertices is randomly reduced by 30%. For smoothing, the edge of the model was smoothed 5 times in CityEditor, but the BER value remains low. Fig.13 shows the original watermark and extracted watermarks under the attacks mentioned above.

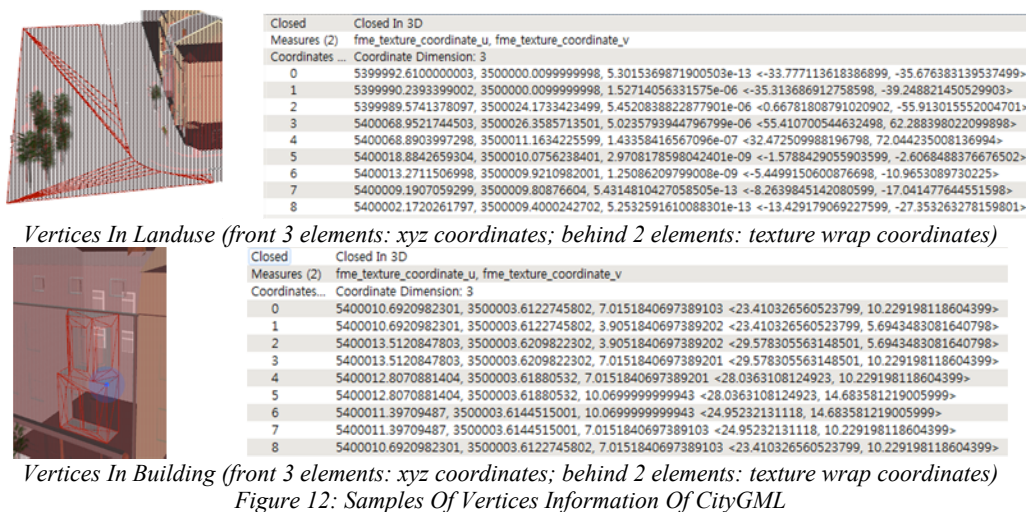


Figure 12: Samples Of Vertices Information Of CityGML

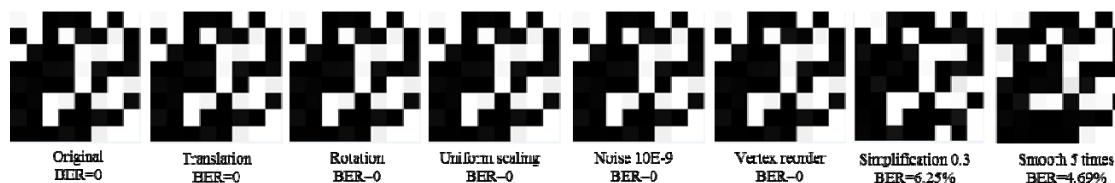


Figure 13: Original Watermark And Extracted Watermarks Under Attacks

## 5. CONCLUSIONS

In this paper, we presented a zero-watermarking method for CityGML model. The method filled in the gaps in the fields of CityGML copyright protection. The global geometry property vertex norm based method can resist several common attacks such as rotation, scaling, transform, vertex reordering, noise addition, simplification, and smoothing. However, the limited numbers of the points in the model object inhibit the capacity and the robustness of the watermark. Considering that each object in CityGML can have a different representation for every LOD, therefore, in the future, we will study the watermarking algorithms for objects with different LODs to enlarge the watermark capacity

and combine much more features of CityGML to design a blind digital watermark.

## ACKNOWLEDGMENTS:

This work was supported by Ministry of Culture, Sports and Tourism (MCST) and Korea Creative Content Agency (KOCCA) in the Culture Technology (CT) Research & Development Program 2017.

## REFERENCES:

- [1] M. Lancelle, and D. W. Fellner, "Current issues on 3D city models", In Proceedings of the Proceedings of the 25th International Conference in Image and Vision Computing, Queenstown, New Zealand, 8-9 November 2010, pp. 363-369



- [2] G. Gröger, T. H. Kolbe, C. Nagel, K. H. Häfele, “OGC City Geography Markup Language (CityGML) Encoding Standard”, <https://www.citygml.org/>, 2012
- [3] G. Gerhard, L. Plümer, “CityGML–Interoperable semantic 3D city models”, *ISPRS Journal of Photogrammetry and Remote Sensing*, vol.71, pp.12-33. (2012).
- [4] H. C. Fan, L. Q. Meng, “A three-step approach of simplifying 3D buildings modeled by CityGML”, *International Journal of Geographical Information Science*. Vol. 26, No. 6, 2012, pp.1091–1107
- [5] G. Marcus, “Towards generating highly detailed 3D CityGML models from OpenStreetMap”, *International Journal of Geographical Information Science*, Vol.27, No.5, 2013, pp. 845-865
- [6] S. U. Baig, and A. A. Rahman. “A Unified Approach for 3D Generalization of Building Models in CityGML”, *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Volume XL-4/W1, 29th Urban Data Management Symposium, 29 – 31 May, 2013, London, United Kingdom.
- [7] F. Biljecki, J. Stoter, H. Ledoux, S. Zlatanova, and A. Çöltekin, “Applications of 3D city models: State of the art review”, *ISPRS International Journal of Geo-Information*, Vol. 4, No.4, 2015, pp. 2842-2889
- [8] S. K. Jusuf, B. Mousseau, G. Godfroid, and V. S. J. Hui, “Integrated modeling of CityGML and IFC for city/neighborhood development for urban microclimates analysis”, *Energy Procedia*, Vol.122, 2017, pp. 145-150.
- [9] A. Abubahia, M. Cocea, “A clustering approach for protecting GIS vector data”, In *International Conference on Advanced Information Systems Engineering*, Springer, Cham, 2015, pp. 133-147
- [10] Z. X. Lin, P. Fei, and L. Min, “A reversible watermarking for authenticating 2D vector graphics based on bionic spider web”, *Signal Processing: Image Communication*, 2017
- [11] J. Liu, Y. Wang, Y. Li, R. Liu, and J. Chen, “A robust and blind 3D watermarking algorithm using multiresolution adaptive parameterization of surface”, *Neurocomputing*, 2017, pp.304-315
- [12] Z. Stefanos, A. Tefas, and I. Pitas, “Blind robust watermarking schemes for copyright protection of 3D mesh objects”, *IEEE Transactions on Visualization and Computer Graphics*, Vol. 11, No. 5, 2005, pp. 596-607
- [13] K. Wang, G. Lavoué, F. Denis, and A. Baskurt, “Robust and blind mesh watermarking based on volume moments”, *Computers & Graphics*, Vol.35, No.1, 2011, pp. 1-19
- [14] X. Y. Wang, and Y. Z. Zhan, “A zero-watermarking scheme for three-dimensional mesh models based on multi-features”, *Multimedia Tools and Applications*, 2017, pp. 1-28.
- [15] M. P. B. Rane, and B. Meshram, “Xml-Based security for e-commerce application”, *International Journal of Scientific and Research Publications*, Vol. 2, No. 7, 2012, pp. 1-4
- [16] T. Romaric, E. Damiani, and N. Bennani, “Robust XML watermarking using fuzzy queries”, In *Computer Software and Applications Conference Workshops (COMPSACW)*, IEEE 36th Annual, 2012, pp. 433-438
- [17] <https://www.3dcitydb.org/3dcitydb/citygml4j/>
- [18] <https://www.citygml.org/samplefiles/>