

# AN IMPROVED LOD FRAMEWORK FOR THE TERRAINS IN 3D CITY MODELS

Kavisha Kumar\*, Anna Labetski, Hugo Ledoux, Jantien Stoter

3D Geoinformation Group, Delft University of Technology, the Netherlands -  
(k.kavisha, a.labetski, h.ledoux, j.e.stoter)@tudelft.nl

Commission IV, WG IV/10

**KEY WORDS:** Level of Detail, LOD framework, terrain, CityGML, 3D city models

## ABSTRACT:

The Level of Detail (LOD) concept in CityGML 2.0 is meant to differentiate the multiple representations of semantic 3D city models. Despite the popularity and general acceptance of the concept by the practitioners and stakeholders in 3D city modelling, there are still some limitations. While the CityGML LOD concept is well defined for buildings, bridges, tunnels, and to some extent for roads, there is no clear definition of LODs for terrain/relief, vegetation, land use, water bodies, and generic city objects in CityGML. In addition, extensive research has been done to refine the LOD concept of CityGML for buildings but little is known on requirements and possibilities to model city object types as terrain at different LODs. To address this gap, we focus in this paper on the terrain of a 3D city model and propose a framework for modelling terrains at different LODs in CityGML. As a proof of concept of our framework, we implemented a software prototype to generate terrain models with other city features integrated (e.g. buildings) at different LODs in CityGML.

## 1. INTRODUCTION

The concept of Level of Detail (LOD) is an important characteristic of a 3D city model (Biljecki et al., 2016b). It refers to the capability of modelling multiple representations of a spatial object at different levels of data quality and complexity for different applications (Danovaro et al., 2006). The concept has been borrowed from the computer graphics domain and accepted in the 3D GIS community without much discussion. One such implementation of the LOD concept can be seen in the international 3D GIS standard CityGML version 2.0 by the OGC (Open Geospatial Consortium).

CityGML defines 5 LODs and supports multiple representations of the same city object in different LODs simultaneously (OGC, 2012). Unlike computer graphics where point density (number of points per m<sup>2</sup>), number of pixels, distance from the camera, etc. are used, the concept of LODs in CityGML is driven by both semantics and geometry (Biljecki et al., 2016b). These 5 LODs have been widely adopted by practitioners and stakeholders in 3D city modelling.

The CityGML LOD concept is well defined for buildings, bridges, tunnels, and to some extent for roads (Beil et al., 2017; Labetski et al., 2018). However, there is no clear definition of LODs for terrain/relief, vegetation, land use, water bodies, and generic city objects in CityGML (Löwner et al., 2013; Benner et al., 2013; Kumar et al., 2016). There are many open issues in the current standard; there is no distinction between the different LODs of a terrain/relief feature at the geometric and semantic level (Kumar et al., 2016) (Figure 1). Further, it is not clear what an LOD4 representation is for features representing vegetation, land use, or water bodies (Benner et al., 2013), given that LOD4 models the indoors which is for buildings, bridges, and tunnels.

There is no widely-accepted LOD paradigm for terrains in 3D city modelling. CityGML 2.0 specifications also state in the terrain module that “for a LOD3 scene it might be sufficient

to use a regular grid in LOD2 with certain higher precision areas defined by ReliefComponents in LOD3” (OGC, 2012). It is not clear what a “regular grid in LOD2” is. What are those “certain higher precision areas in LOD3”, and how are they defined?

Extensive research has been done to refine the LOD concept in CityGML. Currently, it is mostly concerned with buildings. For instance, Löwner et al. (2013) proposed a new LOD concept for CityGML buildings differentiating a *Geometric LOD (GLOD)* and a *Semantic LOD (SLOD)*, separately defined for the interior and exterior of a building. Biljecki et al. (2016b) proposed an improved LOD specification for the exterior geometry of buildings. Similarly, Tang et al. (2018) implemented an indoor LOD concept for the interiors of buildings using the ADE (Application Domain Extension) mechanism of CityGML.

We focus in this paper on the terrain of a 3D city model and propose a framework for modelling terrains at different LODs in CityGML. Our proposal is based on the TIN (Triangulated Irregular Network) representation of the terrains (*TINRelief* in CityGML). We investigate the current status of the LOD concept for terrains in practice, and identify the shortcomings of the current concept for the terrain LODs in CityGML (Section 2). We present our framework for modelling terrains at different LODs in CityGML in Section 3. As a proof of concept of our framework, we implemented a software prototype to generate terrain models with integrated city features (e.g. buildings) at different LODs in CityGML Section 4. We close the article with conclusions and future work in Section 5.

## 2. BACKGROUND

### 2.1 Terrains in 3D city modelling

Terrain<sup>1</sup> (Latin *Terra* meaning Earth) in simple terms refers to the lay of the land described in terms of elevation, slope, or other attributes of the landscape. The terrain is the visible upper part of the surface of the Earth, i.e. the relief. It forms an

\*Corresponding author

<sup>1</sup><https://en.wikipedia.org/wiki/Terrain>

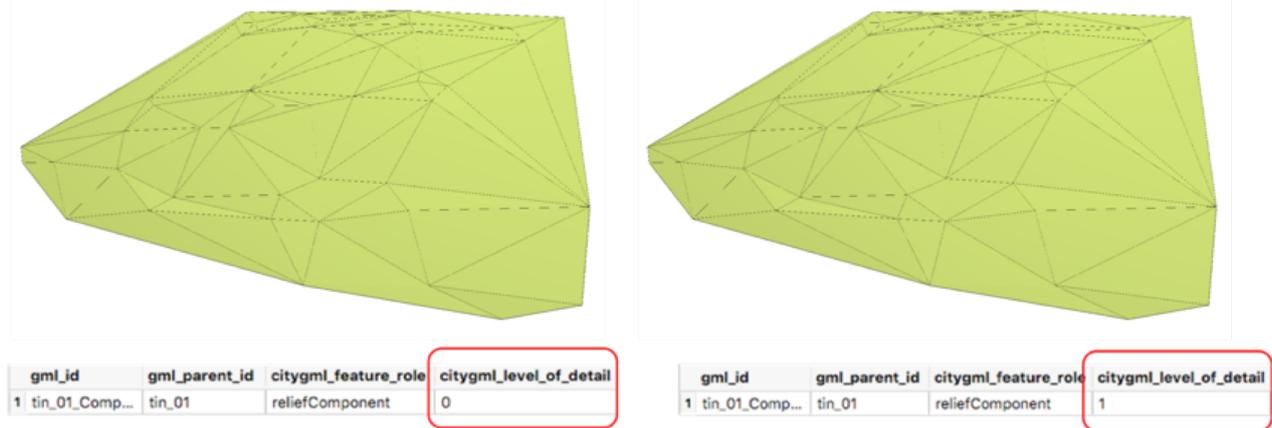


Figure 1. Two CityGML terrains (*TINRelief*) at different LODs, both perfectly valid according to the CityGML description. No distinction between the different LODs of a terrain at geometric and semantic level can be observed.

important part of a 3D city model. Over the last few decades, grids and TINs have become the two most popular models for representing terrains (Kumler, 1994). In this paper we focus on the TIN representation of the terrains for 3D city modelling.

A TIN is a network of non-overlapping triangles formed by the interconnection of irregularly spaced points (Kumler, 1994). For a given set of points, different triangulations can be constructed (Rippa, 1990; Dyn et al., 1990). TINs are generally constructed with Delaunay triangulation to avoid long and skinny triangles (De Berg et al., 2000). But every time the usual inputs for TIN generation are not merely a set of vertices. If the point set is associated with some constraints (segments, polygons, etc.) then a CDT (Constrained Delaunay Triangulation) can be constructed (Figure 2). A CDT is similar to DT but every input segment appears as an edge of the triangulation (Shewchuk, 1996). An example is the 3D TOP10NL (Kadaster, 2015), the 3D city model of the Netherlands, covers the whole country (including buildings, roads, water bodies, and bridges) as one massive triangulation with more than one billion triangles.

TINs are generally 2.5D but they can be more than 2.5D (Gröger et al., 2005; Penninga, 2008). TINs can be modelled to represent features like vertical walls, roof overhangs, caves/tunnels, and overfolds like balconies and dormers (Gröger et al., 2005; Kumar et al., 2018) (Figure 3). A ‘2.5D’ TIN stores only one elevation value (z) for any (x, y) location. However, features such as vertical walls, roof overhangs, caves/tunnels, and overfolds like balconies and dormers cannot be represented with 2.5D TINs. A ‘2.5D+’ TIN stores more than one elevation value (z) for any location (x, y) to model the vertical walls of natural or manmade objects like buildings. A ‘2.75D’ TIN is an extension to ‘2.5D+’ TIN to model overhangs of features such as caves/tunnels, balconies and dormers.

## 2.2 Level of Detail (LOD) modelling for terrains in practice

The concept of LODs originates from the realm of computer graphics where the focus is on balancing between complexity and performance by regulating the amount of detail utilised to represent a virtual world (Luebke et al., 2002). Terrain, specifically, has been examined extensively in the computer graphics domain, mainly in the context of view-dependent level-of-detail control (Hoppe, 1998; Lindstrom et al., 1996). Some approaches focus on pre-computed levels of detail for rendering but many also focus on real-time generation based on statistics

(Xia et al., 1997). There are few definitions for the LODs and it is strictly dictated by geometry. Lindstrom et al. (1996) define consecutive LODs by removing every other column and row of the next higher LOD. Hoppe (1998) focuses instead on building a hierarchy based on edge collapsing and recording their inverses.

LOD in GIS is often linked to the cartographic term *scale*. Scale is often understood in different ways, ranging from *cartographic scale* denoting more detailed information on a map, *geographic scale* denoting the spatial extent of an area, *resolution* denoting the size of the smallest distinguishable part of a spatial data set, and *operational scale* which denotes the scale at which a phenomenon operates (Bian, 1997). Scale became an essential parameter of geospatial datasets and influenced the acquiring, handling, storing, and processing of the data (Goodchild, 2011). At the same time, while there have been many studies conducted around the effect of scale change, or spatial resolution, on analysis, there has not been a formalised approach to defining a unified approach to scale or LOD in the GIS realm (Goodchild, 2011). Furthermore, Biljecki (2017) explains that while there is an association between the terms scale and LOD, with the transition from paper maps the term scale is losing its meaning, and therefore using scale in 3D city modelling should be avoided. Beyond the concept of scale, there are many sources equating LOD to *spatial resolution* in the 2D realm (Budiarto et al., 2009; Wate et al., 2013), but this is also lacking a formalised definition.

## 2.3 Terrain LODs in CityGML

CityGML is an open standard from the OGC for the storage and exchange of 3D city models, including their geometry, semantics, and graphical appearance (OGC, 2012). It is implemented as an application schema of the GML3 (Geography Markup Language version 3.1.1). The data model of CityGML comprises of a core module and several thematic extension modules such as *Building*, *Relief*, *Vegetation*, *LandUse*, *WaterBody*, etc.

The *Relief* module allows for the representation of the terrain as a TIN (*TINRelief*), mass points (*MasspointRelief*), break lines (*BreaklineRelief*), or a grid (*RasterRelief*). It is also possible to represent a terrain with a combination of different terrain types within a single dataset. For instance, a terrain can be modelled by a coarse grid with some areas depicted by detailed TIN or as a TIN with break lines to depict a constrained tri-

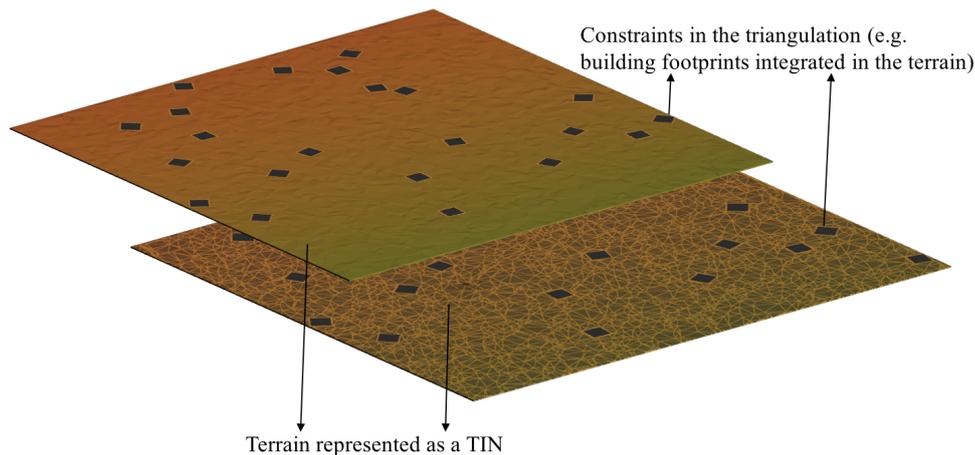


Figure 2. Terrain represented as a constrained TIN with building footprints integrated in it.

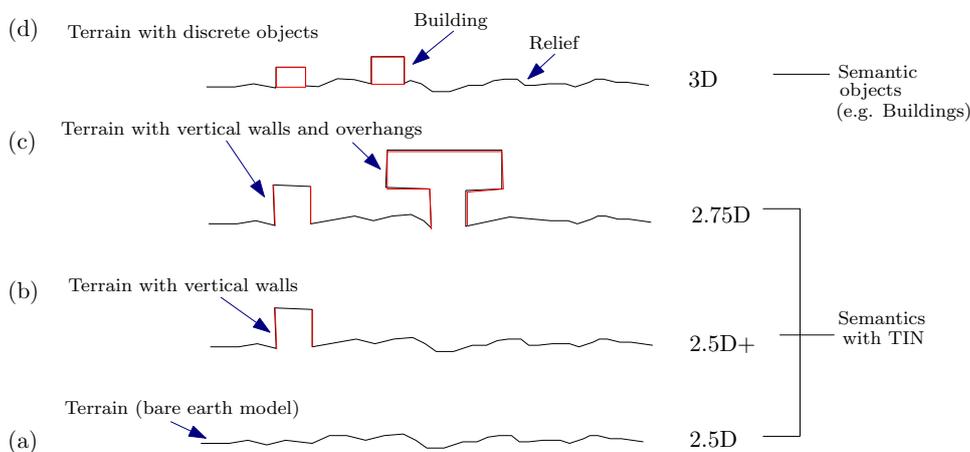


Figure 3. Terrain as a TIN modelled to represent features like vertical walls, roof overhangs, etc. (Source: Kumar et al. (2018))

angulation, etc. A terrain is modelled as a *ReliefFeature* which consists of one or more entities of the class *ReliefComponent*, which can be a TIN or a grid and so on. Both *ReliefFeature* and *ReliefComponent*(s) have an *dem:lod* attribute for the level of detail (OGC, 2012). The LOD of a *ReliefFeature* can differ from the LOD of its *ReliefComponents* (OGC, 2012).

CityGML also supports 5 LODs for the relief. However, there are no guidelines provided that differentiates between these 5 LODs at geometry and semantic level (Kumar et al., 2016). For instance, in Figure 1 the geometry (e.g. number of triangles) of the terrain (*TINRelief*) remains the same though the LOD changes from 0 to 1; the number of points/triangles in every LOD or any other criteria to differentiate between the terrain LODs is not prescribed. The standard has guidance for the LODs of certain modules such as buildings, bridges, and tunnels.

There is also the concept of the Terrain Intersection Curve (TIC) in CityGML. A TIC is used to integrate 3D objects such as buildings with the terrain model. It stores the exact position where a terrain intersects with the 3D objects to avoid the 3D objects float over or sink into the terrain. This is particularly the case if terrains and 3D objects in different LOD are combined, or if they come from different providers (OGC, 2012). However, TICs are seldom used in practice, out of 31 sources of

open 3D city model datasets only 1 source used TICs<sup>2</sup>. Further, the attribute *RelativeToTerrainType* only gives a qualitative reference to the position of a city object with respect to the terrain (*+entirelyAboveTerrain*, *+substantiallyAboveTerrain*, *+substantiallyAboveAndBelowTerrain*, *+substantiallyBelowTerrain*, *+entirelyBelowTerrain*) and not a quantitative measure.

### 3. OUR PROPOSAL FOR MODELLING TERRAIN AT DIFFERENT LODS IN CITYGML

LODs of 3D city models do not differ only by the amount of geometric data, and visual properties, but also they may differ in terms of their semantic information. One geometry based solution to differentiate between TIN terrains at different LODs can be techniques used in computer graphics to restrict the point/triangle density (number of points/triangles per m<sup>2</sup>) required for each LOD, while staying close to the original shape of the terrain. A simplified TIN will have just enough vertices/triangles to model the terrain as per the required level. However, deciding the number of points/triangles for every dataset does not seem feasible. While the number of primitives generally gives a good impression about the geometric complexity of a 3D city model, it cannot be considered as an unambiguous differentiator as is the case in computer graphics. Even in the realm of computer graphics there is no clear consensus about what constitutes a specific LOD.

<sup>2</sup><https://3d.bk.tudelft.nl/opendata/opencities/>

In our proposal, we focus on modelling terrains with respect to the geometry and semantics of the terrain and the integration of terrain with the other city objects present in the city model. The proposal is based on the different TIN representations for modelling terrains considered in Kumar et al. (2018) (see Figure 3). Each succeeding LOD in our framework contains more detail and complexity than the preceding LOD (OGC, 2012; Biljecki et al., 2016b); see Table 1 for the summary of the proposed LODs. We use the CityGML *Generics* module to introduce all the new attributes in our LODs.

### 3.1 LOD0

The terrain in CityGML is a DTM (Digital Terrain Model) and not a DSM (Digital Surface Model). LOD0 is the coarsest and most generalised representation of city objects in CityGML. For terrains (*TINRelief*), we leave LOD0 as a *strict 2.5D TIN representation* (without vertical surfaces and overhangs) i.e. a simple Delaunay triangulation of the ground without man-made objects and vegetation embedded in the TIN (see Figure 4). This ensures that an LOD0 TIN can be readily converted and used in all GIS packages which often assume that a terrain is 2.5D. The terrain is a 2-manifold surface. We also introduce a semantic attribute (*numberOfTriangles*) to store the number of triangles in the TIN (see Snippet 1).

Thus, LOD0 = a strict 2.5D TIN.

```
<dem:ReliefFeature gml:id="relief_feature_01">
  <gml:name>Relief Feature</gml:name>
  <dem:lod>0</dem:lod>
  <dem:reliefComponent>
    <dem:TINRelief gml:id="tin_relief_01">
      <gml:name>TIN model</gml:name>
      <gen:intAttribute name="numberOfTriangles">
        <gen:value>89</gen:value>
      </gen:intAttribute>
      <dem:lod>0</dem:lod>
      <dem:tin>
        <gml:TriangulatedSurface>
          ....
        </gml:TriangulatedSurface>
      </dem:tin>
    </dem:TINRelief>
  </dem:reliefComponent>
</dem:ReliefFeature>
```

Snippet 1: Excerpt of the generated LOD0 terrain model in CityGML

### 3.2 LOD1

The ISO 19107:2003 Spatial Schema (ISO, 2003) standard defines *GM\_TIN* geometry type for representing TIN models, which in theory should allow vertical triangles (surfaces) in a TIN (i.e. more than 2.5D) (Kumar et al., 2018). 3DTOP10NL terrain (TIN) is one such example dataset which has vertical walls (triangles) (Kadaster, 2015). Modelling it in 2.5D will result in the loss of triangles representing the vertical walls. Furthermore, a 2.5D model does not allow for overhangs such as cliffs, naturally-formed arches and caves present in the terrain.

Therefore, we model an LOD1 terrain as an *extension to the 2.5D DTM to support the representation of vertical triangles and overhangs in the TIN i.e. a 2.5D+/.75D model* (see Figure 4). The terrain is still a 2-manifold surface and the software can use and edit it.

We introduce an attribute (*vertTrianglesID*) to store the list of the IDs of these vertical triangles in the model as there is no mechanism in CityGML to flag the vertical triangles. This is important because when a model with vertical triangles is projected on a 2D surface, the vertical surfaces flatten out which distorts the geometry of the model (Kumar et al., 2016). Flagging these vertical triangles allows their removal while transforming from 3D to 2D/2.5D. Similarly, we also introduce an attribute (*ovTrianglesID*) to store the list of the IDs of the triangles representing the overhangs in the model. We also introduce an attribute (*numberOfTriangles*) to store the number of triangles in the TIN.

Thus, LOD1 = LOD0 + information about the vertical triangles and overhangs in the TIN.

```
<dem:ReliefFeature gml:id="relief_feature_01">
  <gml:name>Relief Feature</gml:name>
  <dem:lod>1</dem:lod>
  <dem:reliefComponent>
    <dem:TINRelief gml:id="tin_relief_01">
      <gml:name>TIN model</gml:name>
      <gen:intAttribute name="numberOfTriangles">
        <gen:value>89</gen:value>
      </gen:intAttribute>
      <gen:stringAttribute name="vertTrianglesID">
        <gen:value>vt1 vt2 vt3 ...</gen:value>
      </gen:stringAttribute>
      <gen:stringAttribute name="ovTrianglesID">
        <gen:value>ot1 ot2 ot3 ...</gen:value>
      </gen:stringAttribute>
      <dem:lod>1</dem:lod>
      <dem:tin>
        <gml:TriangulatedSurface>
          ....
        </gml:TriangulatedSurface>
      </dem:tin>
    </dem:TINRelief>
  </dem:reliefComponent>
</dem:ReliefFeature>
```

Snippet 2: Excerpt of the generated LOD1 terrain model in CityGML

### 3.3 LOD2

LOD0 and LOD1 terrain models can be useful in applications such as hydrological flow modelling, natural hazard modelling, geomorphological mapping, and relief maps. However, they cannot be used in applications where information about the location of city objects with respect to the terrain is required e.g. to determine the effect of surface features such as buildings and vegetation on visibility analysis and viewshed calculations (Kidner et al., 2000), hydrological modelling in urban environments to identify the flood risk (Gorte et al., 2012), etc.

Therefore, we model an LOD2 terrain as a *semantically enriched strict 2.5D DTM* with information about the city objects integrated in the terrain. For this, we define an LOD2 terrain as a constrained Delaunay triangulation where the boundaries of the city objects such as buildings, roads, etc. act as constraints in the triangulation (see Figure 4).

We introduce an attribute to store the extent/boundary (*extent*) of the triangles representing the footprints of city objects in the terrain. Further attributes are added to store information about the type of city object (*cityObjectType*) and the ID of the city

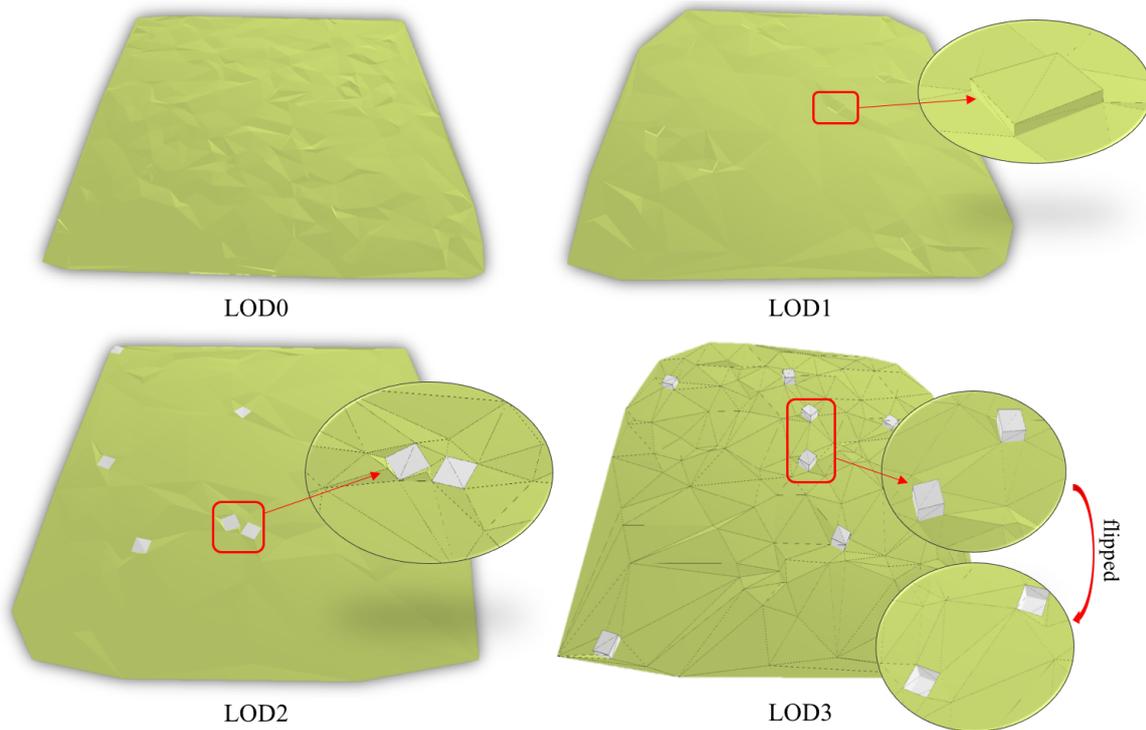


Figure 4. Terrain models at different LODs (0-3) according to our framework. (Source: Generated by our *Random3DTerrain* software prototype.)

object (*cityObjectID*) represented by these triangles (see Snippet 3). We also introduce an attribute (*numberOfTriangles*) to store the number of triangles in the TIN.

Thus, LOD2 = LOD0 + semantic information about the city objects integrated in the terrain.

The same LOD2 terrain can integrate with both, the LOD0 building footprints and the LOD1 block model of the buildings because the building footprints remain the same for LOD0 and LOD1 buildings. It can also fit with higher LOD models of buildings provided their footprints (ground surface) remain the same. If their footprint change, then re-computation of the TIN is required.

```

<dem:ReliefFeature gml:id="relief_feature_01">
  <gml:name>Relief Feature</gml:name>
  <dem:lod>2</dem:lod>
  <dem:reliefComponent>
    <dem:TINRelief gml:id="tin_relief_01">
      <gml:name>TIN model</gml:name>
      <gen:intAttribute name="numberOfTriangles">
        <gen:value>89</gen:value>
      </gen:intAttribute>
      <gen:genericAttributeSet name="Fprint_b01">
        <gen:stringAttribute name="cityObjectType">
          <gen:value>Building</gen:value>
        </gen:stringAttribute>
        <gen:stringAttribute name="cityObjectID">
          <gen:value>b01</gen:value>
        </gen:stringAttribute>
        <gen:stringAttribute name="extent">
          <gen:value>155067.7614 466489.4299 21.27
            ....</gen:value>
        </gen:stringAttribute>
      </gen:genericAttributeSet>
      <gen:genericAttributeSet name="Fprint_b02">
        ....
      </gen:genericAttributeSet>
    </dem:TINRelief>
  </dem:reliefComponent>
</dem:ReliefFeature>
    
```

Snippet 3: Excerpt of the generated LOD2 terrain model in CityGML

### 3.4 LOD3

Lastly, we model an LOD3 terrain as a *semantically enriched 2.5D+2.75D extended DTM* with information about the city objects integrated in the terrain (see Snippet 4 and Figure 4). In short,

LOD3 = LOD1 + semantic information about the city objects integrated in the terrain.

We introduce two attribute *vertTrianglesID* and *ovTrianglesID* to store the list of the IDs of these vertical triangles and triangles representing the overhangs in the model. We also introduce an attribute (*numberOfTriangles*) to store the number of triangles in the TIN. Other attributes are added to store the type of city object (*cityObjectType*) and the ID of the city object (*cityObjectID*) present in the dataset.

```
<dem:ReliefFeature gml:id="relief_feature_01">
  <gml:name>Relief Feature</gml:name>
  <dem:lod>3</dem:lod>
  <dem:reliefComponent>
    <dem:TINRelief gml:id="tin_relief_01">
      <gml:name>TIN model</gml:name>
      <gen:intAttribute name="numberOfTriangles">
        <gen:value>89</gen:value>
      </gen:intAttribute>
      <gen:stringAttribute name="vertTrianglesID">
        <gen:value>vt1 vt2 vt3 ...</gen:value>
      </gen:stringAttribute>
      <gen:stringAttribute name="ovTrianglesID">
        <gen:value>ot1 ot2 ot3 ...</gen:value>
      </gen:stringAttribute>
      <gen:genericAttributeSet name="Fprint_b01">
        <gen:stringAttribute name="cityObjectType">
          <gen:value>Building</gen:value>
        </gen:stringAttribute>
        <gen:stringAttribute name="cityObjectID">
          <gen:value>b01</gen:value>
        </gen:stringAttribute>
        <gen:stringAttribute name="extent">
          <gen:value>155067.7614 466489.4299 21.27
            ....</gen:value>
        </gen:stringAttribute>
      </gen:genericAttributeSet>
      <gen:genericAttributeSet name="Fprint_b02">
        ....
      </gen:genericAttributeSet>
    </dem:TINRelief>
  </dem:reliefComponent>
</dem:ReliefFeature>
```

Snippet 4: Excerpt of the generated LOD3 terrain model in CityGML

### 3.5 LOD4 = removed

We do not define an LOD4 representation for the terrains for the following reasons:

1. LOD4, in general, models the interior of city objects such as buildings, tunnels, bridges, etc., this does not make sense in relation to terrains.
2. CityGML 3.0, the upcoming version of CityGML, plans to phase out the LOD4 representation of features.

#	LOD	Description
1	LOD0	LOD0 = a strict 2.5D TIN representation.
2	LOD1	LOD1 = LOD0 + information about the vertical triangles and overhangs in the TIN.
3	LOD2	LOD2 = LOD0 + information about the city objects integrated in the terrain.
4	LOD3	LOD3 = LOD1 + information about the city objects integrated in the terrain.

Table 1. Summary of the proposed terrain LODs

## 4. IMPLEMENTATION

In order to test our proposed framework and show its usability, we developed a software prototype, *Random3DTIN* which generates artificial TIN terrain models at different LODs (0-3) in CityGML format (Figure 4). Our prototype is based on the procedural modelling engine *Random3DCity*<sup>3</sup> developed by [Biljecki et al. \(2016a\)](#) for generating random CityGML buildings in multiple LODs.

The new attributes for the terrain e.g. number of triangles, ID and type of the city object, etc. are introduced as *Generic* attributes in the generated CityGML datasets (see Snippets 1, 2, 3 and 4). The software we developed, together with the sample datasets, is freely available in our GitHub repository: <https://github.com/tudelft3d/Random3DTIN>.

## 5. CONCLUSION AND FUTURE WORK

The concept of Level of Detail in CityGML is widely used by practitioners and stakeholders in the field of 3D city modelling for representing city features with varying degrees of complexity in the geometry and semantics as per the need of a specific application. Nevertheless, it lacks a precise definition of each LOD for features such as terrain, water body, vegetation, and land use.

In this paper we presented our framework for modelling terrains at different LODs in CityGML. There is currently no distinction between the different LODs of a terrain/relief at the geometric and semantic level in CityGML. The framework that we propose is simple and compliant with the existing LOD concept in CityGML and is meant to improve the ambiguity of the current concept. It also makes an explicit distinction between 2.5D and more complex representations of terrain, given that many GIS softwares only support 2.5D. Therefore, being able to enforce 2.5D in LOD0 and LOD3 is useful. Moreover, our approach can also be extended to CityJSON<sup>4</sup>, which is a JSON encoding for a subset of the CityGML data model. The methodology does not restrict the LODs to the geometric data granularity in values such as the number of points/triangles; these values are often arbitrary or application/user specific. Rather, our approach aims to integrate the terrain with surrounding features while adding further geometric and semantic information. We further aim to undertake a follow-up study to iterate and validate the model-

<sup>3</sup><https://github.com/tudelft3d/Random3DCity>

<sup>4</sup><https://www.cityjson.org>

ling choices made for each LOD for use in different applications.

As a proof of concept, we also developed a software prototype (*Random3DTIN*) to generate CityGML terrain datasets with other integrated city features, such as buildings, based on our framework. The prototype is open source and a set of sample datasets is available for free, for public use, so that other researchers can benefit from the different LOD terrain datasets in their application domains. The prototype generates artificial terrain models suited for applications where having real world data is not important such as in the case of testing simulations or analysis to determine which LOD is best suited for the process. The open source 3D city modelling software *3dfier*<sup>5</sup> generates real world terrain models with vertical walls, overhangs, etc. The output from *3dfier* can easily be adjusted to represent these terrain models according to our proposed LOD framework.

Our methodology also maintains the modular approach of CityGML by not linking the LODs of the terrain directly to the LODs of the city objects present in a dataset. This means that an LOD2 terrain is not tied to the LOD2 building model or any other LOD2 city objects. Any of the terrain LODs (i.e. LOD0, LOD1, LOD2 and LOD3) can also exist with LOD0, LOD1 or higher LOD city objects. Further, our methodology allows for the storage of the triangles and the triangulation constraints explicitly in the data structure so that same triangulation is enforced.

Compared with the aforementioned *Terrain Intersection Curve* currently present in CityGML, our proposal has several advantages. First, the TIC was designed to assist with the integration of city objects with their surrounding terrain but it is currently not applicable to all city features and only focuses on buildings and building parts, bridge, bridge parts and bridge construction elements, tunnel and tunnel parts, city furniture objects, and generic city objects (OGC, 2012). This excludes transportation objects such as roads and railways, vegetation objects and water bodies, whereas our framework covers all the city objects integrated in the terrain. The knowledge of where the road intersects with the terrain can aid users in calculating more accurate calculations of road inclination. Second, the TIC only defines the geometry of the intersection and has no other semantic information that can be stored, whereas we introduced attributes with semantic information about such intersections in the TIN. 3D road networks can be utilised for optimising routing network for waste collection and transportation (Tavares et al., 2009). Understanding the affect that road inclination and vehicle weight have can aid in optimising for minimum fuel consumption which can result in lower costs than traditional shortest route approaches (Tavares et al., 2009). Our proposal for LOD2 would enable such calculations to be done with CityGML datasets. Last, a TIC is only relevant in context with a terrain, therefore it makes more sense to store intersection information with the terrain and not individual city features.

Generalisation is the process of transitioning down from higher levels of detail to lower levels of detail. Compared to the well-researched world of 2D (cartographic) generalisation, 3D city model generalisation is different in that its primary focus is not visualisation but is rather application-driven (Guercke, Brenner). In the future, we will study how our proposed LOD refinement can assist with the harmonious generalisation of city objects, i.e. generalising multiple city features alongside each

other. The constrained nature of the LOD definition can guide practitioners in ensuring that city objects fit with the terrain after a generalisation process.

It is also necessary to examine the LOD concept for the other terrain representation types. It is not reasonable to have an LOD definition that would be applicable to the entire relief module and therefore there needs to be an investigation into the needs of all representation types separately.

It is also possible to implement the terrain LOD framework as a CityGML ADE, both UML (Unified Modelling Language) and XSD (XML Schema Definition). Further, we are working on refining our prototype to implement terrain tiling and automatically enrich the terrain features in existing 3D city models with our framework.

## ACKNOWLEDGEMENTS

The research leading to this paper is a part of the research project 3D4EM (3D for Environmental Modelling) in the Maps4-Society programme (Grant No. 13740) which is funded by the NWO (Netherlands Organisation for Scientific Research), and partly funded by the Ministry of Economic Affairs. This work is also funded by the European Research Council under the European Union's Horizon 2020 ERC Agreement no. 677312 UMnD: Urban modelling in higher dimensions.

## REFERENCES

- Beil, C., Kolbe, T. H., , 2017. CityGML and the streets of New York-A proposal for detailed street space modelling. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-4/W5*, 9–16.
- Benner, J., Geiger, A., Gröger, G., Häfele, K.-H., Löwner, M.-O., 2013. Enhanced LOD concepts for virtual 3D city models. *ISPRS annals of the photogrammetry, remote sensing and spatial information sciences, II-2/w1*, 51–61.
- Bian, L., 1997. Multiscale nature of spatial data is scaling up environmental models. In 'Scale in Remote Sensing and GIS'. (Eds MF Goodchild, DA Quattrochi) pp. 13–25.
- Biljecki, F., 2017. Level of detail in 3D city models. PhD thesis, Delft University of Technology, Delft, the Netherlands.
- Biljecki, F., Ledoux, H., Stoter, J., 2016a. Generation of multi-LOD 3D city models in CityGML with the procedural modelling engine Random3Dcity. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-4/W1*, 51–59.
- Biljecki, F., Ledoux, H., Stoter, J., 2016b. An improved LOD specification for 3D building models. *Computers, Environment and Urban Systems*, 59, 25–37.
- Budiarto, R., Isawasan, P., Aziz, M. A., 2009. Transformation of spatial data format for interoperability between gis applications. *2009 Sixth International Conference on Computer Graphics, Imaging and Visualization, IEEE*, 536–539.
- Danovaro, E., De Floriani, L., Magillo, P., Puppo, E., Sobrero, D., 2006. Level-of-detail for data analysis and exploration: a historical overview and some new perspectives. *Computers & Graphics*, 30(3), 334–344.

<sup>5</sup><https://github.com/tudelft3d/3dfier>

- De Berg, M., Van Kreveld, M., Overmars, M., Schwarzkopf, O. C., 2000. Computational geometry.
- Dyn, N., Levin, D., Rippa, S., 1990. Data dependent triangulations for piecewise linear interpolation. *IMA journal of numerical analysis*, 10(1), 137–154.
- Goodchild, M. F., 2011. Scale in GIS: An overview. *Geomorphology*, 130(1-2), 5–9.
- Gorte, B., Lesparre, J., 2012. Representation and reconstruction of triangular irregular networks with vertical walls. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVIII-4/C26, 15-19.
- Gröger, G., Plümer, L., 2005. How to get 3-D for the price of 2-D – topology and consistency of 3-D urban GIS. *Geoinformatica*, 9(2), 139–158.
- Guercke, R., Brenner, C., 2009. A framework for the generalization of 3d city models. *Proceedings of 12th AGILE Conference on GIScience*, 31.
- Hoppe, H., 1998. Smooth view-dependent level-of-detail control and its application to terrain rendering. *Proceedings Visualization'98 (Cat. No. 98CB36276)*, IEEE, 35–42.
- ISO, 2003. 19107: 2003(E) Geographic information: Spatial schema.
- Kadaster, 2015. 3DTOP10NL. <http://arcg.is/1GKYy7E>. Accessed 05 May 2019.
- Kidner, D. B., Ware, J. M., Sparkes, A. J., Jones, C. B., 2000. Multiscale terrain and topographic modelling with the implicit TIN. *Transactions in GIS*, 4(4), 361–378.
- Kumar, K., Ledoux, H., Stoter, J., 2016. A CityGML extension for handling very large TINs. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2/W1, 137-143.
- Kumar, K., Ledoux, H., Stoter, J., 2018. Compactly representing massive terrain models as TINs in CityGML. *Transaction in GIS*, 22(5), 1152–1178. <https://doi.org/10.1111/tgis.12456>.
- Kumler, M., 1994. An intensive comparison of triangulated irregular networks (TINs) and digital elevation models (DEMs). *Cartographica*, 31(2), 1.
- Labetski, A., van Gerwen, S., Tamminga, G., Ledoux, H., Stoter, J., 2018. A Proposal for an Improved Transportation Model in CityGML. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-4/W10, 89–96.
- Lindstrom, P., Koller, D., Ribarsky, W., Hodges, L. F., Faust, N. L., Turner, G., 1996. Real-time, continuous level of detail rendering of height fields. Technical report, Georgia Institute of Technology.
- Löwner, M.-O., Benner, J., Gröger, G., Häfele, K.-H., 2013. New concepts for structuring 3d city models—an extended level of detail concept for citygml buildings. *International Conference on Computational Science and Its Applications*, Springer, 466–480.
- Luebke, D., Watson, B., Cohen, J. D., Reddy, M., Varshney, A., 2002. *Level of Detail for 3D Graphics*. Elsevier Science Inc., New York, NY, USA.
- OGC, 2012. OGC City Geography Markup Language (CityGML) Encoding Standard 2.0.0. Doc. No. 12019. [https://portal.opengeospatial.org/files/?artifact\\_id=47842](https://portal.opengeospatial.org/files/?artifact_id=47842).
- Penninga, F., 2008. 3D topography: a simplicial complex-based solution in a spatial DBMS. PhD thesis, TU Delft, Delft University of Technology.
- Rippa, S., 1990. Minimal roughness property of the Delaunay triangulation. *Computer Aided Geometric Design*, 7(6), 489–497.
- Shewchuk, J. R., 1996. Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. *Applied Computational Geometry: Towards Geometric Engineering*, 1148, 203–222. <http://www.cs.cmu.edu/quake/triangle.html>.
- Tang, L., Li, L., Ying, S., Lei, Y., 2018. A Full Level-of-Detail Specification for 3D Building Models Combining Indoor and Outdoor Scenes. *ISPRS International Journal of Geo-Information*, 7(11), 419.
- Tavares, G., Zsigraiova, Z., Semiao, V., Carvalho, M. d. G., 2009. Optimisation of MSW collection routes for minimum fuel consumption using 3D GIS modelling. *Waste Management*, 29(3), 1176–1185.
- Wate, P., Srivastav, S., Saran, S., Murthy, Y. K., 2013. Formulation of hierarchical framework for 3d-gis data acquisition techniques in context of level-of-detail (lod). *2013 IEEE Second International Conference on Image Information Processing (ICIIP-2013)*, IEEE, 154–159.
- Xia, J. C., El-Sana, J., Varshney, A., 1997. Adaptive real-time level-of-detail based rendering for polygonal models. *IEEE Transactions on Visualization and Computer graphics*, 3(2), 171–183.