3D GEOSPATIAL DATABASE SCHEMA FOR ISTANBUL 3D CITY MODEL

Alias Abdul Rahman[1], Hanis Rashidan[1], Ivin A. Musliman[1],
Gurcan Buyuksalih[2], Serdar Bayburt[2], and Peyami Baskaraca[2]

[1]3D GIS Research Lab, Universiti Teknologi Malaysia, Johor Bahru, Johor, Malaysia
alias@utm.my; muhd_hanis@yahoo.com; ivinamri@utm.my
[2]BIMTAS, Istanbul, Turkey
{gurcan.buyuksalih, serdar.bayburt, apeyami.baskaraca}@bimtas.istanbul

KEY WORDS*:* 3D city models, PostGIS, CityGML, 3D database, Web Feature Service and Cesium

ABSTRACT:

This paper describes the development of 3D database for Istanbul 3D city models. The schema is based on object-relational technology and also called spatially-enhanced relational database management system (SRDBMS). The 3D city models were generated based on LiDAR points cloud with other several typical GIS datasets like terrain, orthophoto, Point of Interests (POIs), and other attribute data. The database is based on PostGIS schema and CityGML schema (3DCityDB). Two major datasets, namely, terrain data (with several formats), and 3D city models were populated in the database. Terrain and attributes data retrieval are based on Web Feature Service (WFS) whereas 3D models were visualized via 3D Tiles format in Cesium platform. Two issues were also highlighted in the paper with respect to 3D attributes linkages and 3D complex objects.

1. INTRODUCTION

3D city model are being developed by municipalities of several major cities around the globe. The development of the models due to demand by users and citizens with respect to information and services from the city authorities. In general, 3D information able to provide more knowledge of the area of interest (Abdul-Rahman and Pilouk, 2009). The needs for 3D city models are growing and expanding rapidly in a variety of fields. In a steady shift from traditional 2D GIS toward 3D GIS, a great amount of accurate 3D city models has become necessary to be produced in a short period of time and provided widely on the market.

The Istanbul Greater Municipality has developed 3D city models by using 3D point clouds data, 2D digital map, and 3D laser scanned datasets, which are processed by some commercial software and in-house tools. Several tools are available for texture provides photo-realistic and accurate virtual urban space for a variety of uses including 3D-GIS, urban planning, design and redevelopment, hazard prevention, broadcasting, and navigation (Takase et al, 2011). The 3D city models applications are largely investigated research topic and have favour toward a Web-based frameworks solution (El Haje et al., 2016). Thus, it would be more meaningful if it comes with 3D database solution as reported by Yao et al (2018). The development of 3D database forms major discussions of this paper.

The remaining of this paper discusses 3D city modelling in Section 2, the 3D database development in Section 3, and finally conclusions and outlook of the presented work in Section 4.

2. 3D CITY MODELLING AND CityGML

The Istanbul 3D city model has been developed from several data sources, namely, airborne LiDAR, mobile LiDAR and other auxiliary data. The model covers 5400 km^2 of Istanbul city. The airborne LiDAR data were collected for the city of Istanbul with Riegl Q680i scanner during the years 2012 and 2014. The flight campaign was carried out at an average flying height of 500 m by a helicopter at 80 knot flight speed. In total, approximately 17,000 LAS files with the file size of 500-meter by 700-meter have been generated for the whole city.

The main object classes such as ground, building, vegetation (low, medium, high) were derived from these LAS files using the macros in TerraSolid software. After quality check for the classified point cloud, the 3D building cubes (approximately 1.5 million) were obtained automatically using TerraModeller software as a 3D CAD files with the contributions of ortophotos and building footprints available in the 1/1000 scale base maps while some specific structures such as mosques, and bridges which have complex roof type and fancy structures were manually generated. Then, these files were converted to the CityGML format using an FME Workbench. Finally, CityGRID software was used to generate LoD2 and LoD3 models of the buildings as described in Peyami et al (2019).

City Geography Markup Language (CityGML) is an international standard for the interoperable representation and exchange of virtual 3D city and landscape models. CityGML defines a conceptual schema for the most relevant entities of the urban space like buildings, roads, railways, tunnels, bridges, city furniture, water bodies, vegetation, and the terrain. The conceptual schema specifies how and into which parts and pieces physical objects of the real world should be decomposed and classified.

All objects can be represented with respect to their semantics, 3D geometry, 3D topology, and appearances in five predefined levels of detail (LoD 0 to 4). CityGML is formally specified using UML class diagrams, explanations of the object classes and attributes, and an XML schema for the file exchange format. CityGML is issued by the Open Geospatial Consortium (OGC) and the first official version of CityGML was released in

https://doi.org/10.5194/isprs-archives-XLII-4-W16-11-2019 | © Authors 2019. CC BY 4.0 License.

the year 2008 and the current version 2.0.0 was published in 2012 as reported by Yao et al (2018). Several classes and datatypes are categorised into a number of thematic modules and represented in UML diagram.

Since CityGML is an application schema for GML, all geometries supported by GML are supported by CityGML with one exception: while GML allows the use of non-linear geometries, CityGML uses linear ones only. Areal features are represented by triangles and polygons, while volumetric geometries are represented with a boundary representation scheme (b-rep) using triangles/polygons. CityGML presents an appealing solution for the storage and exchange of 3D city models because it combines geometry and semantics in a single data model. However, efficiently visualising 3D geometries and semantic information stored in CityGML is complex (Arroyo et al, 2018).

3. DESIGN AND IMPLEMENTATION OF 3D DATABASE

The increasing complexity and large amount of building data for 3D city modelling requires the efficient management mechanism of spatial data storage. The traditional Relational Database Management System (RDBMS) has several weaknesses in managing 3D geometry and attribute data due to the incompleteness of building data in 2D spatial database (Yang, et al., 2000). Modern 3D city modelling (for urban management and toward smart cities) has to deal with complex requirements and maintain more complex spatial data which requires robust 3D data structure to support efficient data visualization, query and analysis.

This is due to 3D spatial information relies heavily on the database, the component that holds important information of the area. If the component were properly designed then various questions or queries could be performed thus right decisions from decision makers. That is the standard scenario for any geoinformation system either in 2D or 3D as recently reported by Kolbe (2015). As mentioned earlier, this paper is to develop a database for spatial objects within 3D city model of Istanbul that based on LiDAR point clouds. It is a very important component of any geospatial based information system including 3D city model that is to provide queries and answers to any questions related to any objects within the city.

Management, analysis, and visualization of the CityGML based objects could be made possible with this spatial database. Previously, all objects were stored as flat files in a server. Proper geospatial database is the solution as illustrated in Figure 1. One of the options is via 3DCityDB web client (open source) within PostgresSQL that utilises CityGML schema definition files and Cesium Virtual Globe. 3DCityDB is CityGML compliant. Figure 1 also shows the proposed system architecture for better 3D spatial data management, potential application platforms and our proposed Web applications.

3.1 Database Schema

3D city models are nowadays used as an integrative information backbone representing relevant urban entities along with their spatial, semantic, and visual properties. They are often created and maintained with full coverage of entire cities, i.e. buildings, roads, trees (including vegetation), water bodies, and terrains. In order to establish a common understanding and interpretation of the urban objects and to achieve interoperable access and exchange of complete 3D models including the geometric, topologic, visual, and semantic data, the Open Geospatial Consortium (OGC) has issued the CityGML standard (Gröger et al., 2012). CityGML defines a feature catalogue and data model for the most relevant 3D topographic elements (the objects) as mentioned above. The data model is mapped to an XML-based exchange format using OGC's Geography Markup Language (GML).

Figure 1 illustrates CityGML data schemas and modules that represent spatial properties of its thematic classes. As for version 2.0 of the CityGML standard, there are modules for *Appearance, Bridge, Building, CityFurniture, CityObjectGroup, Generics, LandUse, Relief, Transportation, Tunnel, Vegetation,* and *WaterBody.* Each module is represented by a package. A dashed arrow in the figure indicates that the schema at the tail of the arrow depends upon the schema at the head of the arrow.

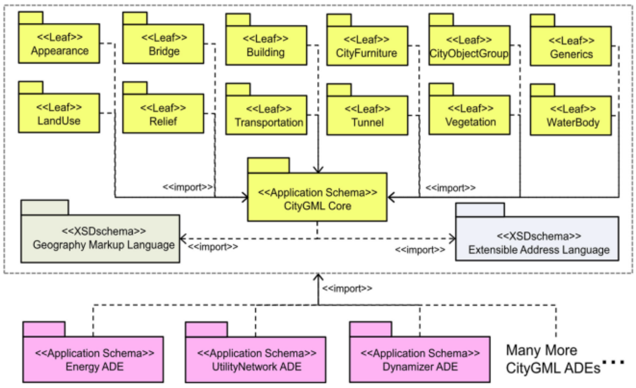


Figure 1. UML package diagram illustrating the separate modules of CityGML and their schema dependencies. (Source: OGC CityGML Encoding Standard, 2012)

3DCityDB has been utilised in this study – an open source package consisting of a database schema and a set of software tools to import, manage, analyse, visualize, and export virtual 3D city models according to the CityGML standard. The database schema implements the CityGML standard with semantically rich and multi-scale urban objects facilitating complex analysis tasks, far beyond visualization. It has been used in real life systems production in several cities and being used in a number of research projects. For example, the cities of Berlin, Potsdam, Munich, Frankfurt, Zurich, Rotterdam, Singapore all keep and manage their virtual 3D city models within an instance of 3DCityDB.

The 3D City Database comes with tools for data exchange and its content can be directly exported in KML, COLLADA, and glTF formats for the visualisation in a broad range of applications like Cesium Virtual Globe, Google Earth, and ArcGIS (Yao, et at., 2018).

In order to setup new database for Istanbul 3D City Models, PostgreSQL (also referred to as Postgres) has been utilised in the study. It is an open-source relational database management system (RDBMS) emphasizing extensibility and standards compliance with capability to handle workloads ranging from single-machine applications to Web services with many concurrent users. The installed database engine comes with an extension called PostGIS. PostGIS is an open source software program that adds support for geographic objects to the

PostgreSQL object-relational database. PostGIS follows the Simple Features for SQL specification from the Open Geospatial Consortium (OGC). It adds support for geographic objects allowing location queries to be run in SQL.

In addition to basic location awareness, PostGIS offers many features rarely found in other competing spatial databases such as Oracle Locator/Spatial and SQL Server (PostGIS.net, 2019). PostGIS adds extra types (such as geometry and raster) to the PostgreSQL database. It also adds functions, operators, and index enhancements that apply to the spatial data types, thus making it a fast, feature-plenty, and robust spatial database management system. Appendix A illustrates the database schema of the Istanbul 3D city models based on CityGML standard that are applied in the PostgreSQL database.

Besides 3DCityDB, other database solution that supports the management of CityGML data is GeoRocket. There are several studies that have been carried out on Georocket as a database solution. One of the studies such as by Koukofikis, Coors, & Gutbell (2018), they conducted an investigation on the interoperable visualization of 3D city models for rendering of hierarchical organized massive 3D data (CityGML) in various web client which are requested via the 3D Portrayal Service queries. Santhanavanich et al, (2018) studied on the integration and visualization of heterogeneous sensor data and geospatial information has utilised Georocket GeoToolbox tool to convert the data into Cesium 3D Tile format for visualisation purposes. GeoRocket is a high-performance data store for geospatial files and focus on the storage, indexing, and analysis of big vector data. It supports CityGML (3D city models), GeoJSON, GML or any other XML-based geospatial data format. It provides scalable high-performance data storage with multiple back-ends such as Amazon S3, MongoDB, H2 (default), or distributed file systems (e.g. HDFS or Ceph). GeoRocket exists in two editions – an Open-Source and a Pro edition for enterprise applications. Figure 2 illustrates the data flow process from GeoRocket to Cesium via 3D Portrayal Services.

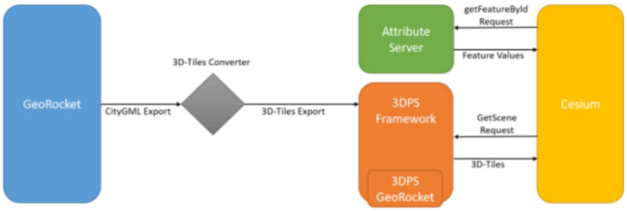


Figure 2. The data-flow from GeoRocket to the visualized 3D Tiles, which are requested via the 3D Portrayal Service queries (source: Koukofikis et al, 2018)

3.2 Database Population

There are several types of data (with respective formats) in the development of Istanbul 3D city models database such as – 3D Models *(CityGML)*; PointAddress *(shapefile point)*; Road *(shapefile line);* Footprint *(shapefile polygon)*; Neighborhood *(shapefile polygon)*; District *(shapefile polygon)* and basemap *(raster)*. The datasets are based in Küçükçekmece, one of the districts in Istanbul. The data originally in flat-file formats and undergone several processes of data validation (coordinate system, geometry, topology, and attributes). Finally, the data are projected into *WGS 1984 UTM Zone 35N 2 (meters)* and imported into the database named *citydb_istanbul*. Figure 3 shows the process of data reprojection of vectors and 3D models into the specified coordinate system.

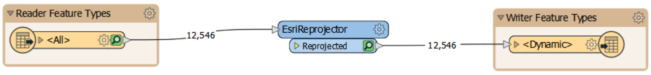
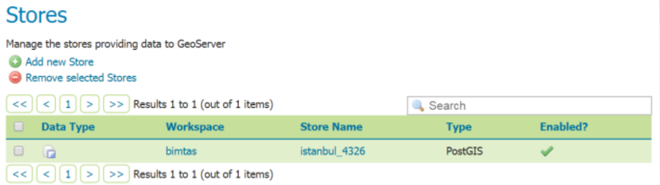


Figure 3. Data reprojection of vectors and 3D models

3.3 Web Feature Service

Geoserver is used to setup Web services for the development of Istanbul 3D City Models. It is known as an open-source server that allows users to share, process and edit geospatial data. It has been designed for interoperability, and publishes data from any major spatial data source using open standards. Figure 4 shows the Geoserver admin interface used in this project for managing, publishing and controlling data layers. Geoserver can directly connect to PostgreSQL database server for WFS and WMS data publishing and provides several operations to manipulate information of geographic features. These operations allow to execute transactions (query, create, update or delete) on data layers through the Web.



Stores

Manage the stores providing data to GeoServer

Add new Store

Remove selected Stores

<< < 1 > >> Results 1 to 1 (out of 1 items)

Search

□	Data Type	Workspace	Store Name	Type	Enabled?
□		bimtas	istanbul_4326	PostGIS	✓

<< < 1 > >> Results 1 to 1 (out of 1 items)

Figure 4. GeoServer publishing spatial data layers from the database through a PostGIS connection.

3.4 Database Query

Queries to the database can be achieved through WFS, such as *GetFeature* operation. The Web client should be able to specify which feature properties to fetch and should be able to constrain the query spatially and non-spatially. The *GetFeature* operation will returns a selection of features from a data source including geometry and attribute values. Below is the example syntax used to get the feature(s) attributes – from 3D Web client to the database via WFS.

http://localhost:8080/geoserver/wms?service=WFS&version=1.1.1&request=GetFeature&typeName=bimtas:footprint&maxFeatures=200&outputformat=application/json&cql_filter=(yapi_id=%27105125336%27)

Then, the server will return the query as illustrated in Figure 6 (in info box).

3.5..3D Visualization

Figure 5 illustrates five major components in the development of Istanbul 3D city models. They are *Database Server* – hosting and manage 2D and 3D spatial datasets, *Web Services* – publish and handling data request from the Web sockets, *3D Web Client* – a platform to visualise and interact with the 3D models (with query), *Desktop GIS* – direct data access, update and edit to the database, and *Existing Datasets* – new or existing datasets.

Figure 6 shows the visualisation of the 3D models within 3D Web client Cesium based on 3D Tiles technique. The visualisation of CityGML models on the web is however still a challenging area since CityGML is designed for the representation of 3D city models and not for presenting or visualising the 3D city models directly on web (Arroyo et al, 2018).

https://doi.org/10.5194/isprs-archives-XLII-4-W16-11-2019 | © Authors 2019. CC BY 4.0 License.

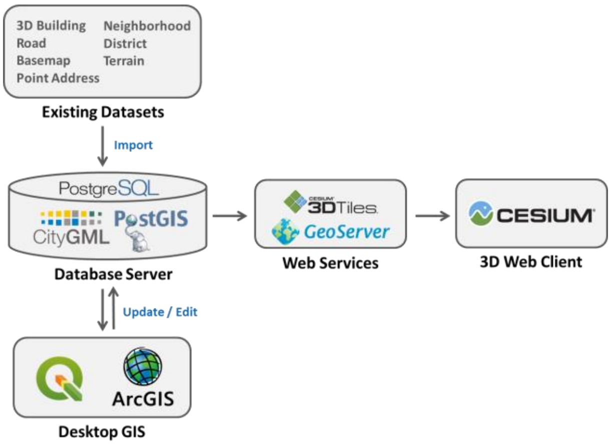
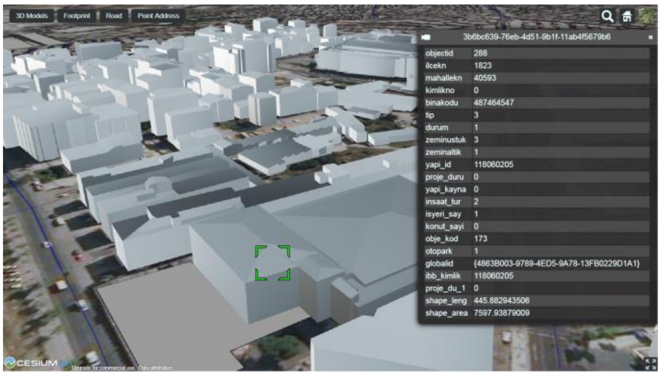


Figure 5. An integrated schematic diagram of the data flow (import, export, update, edit, retrieval, access) and the associated database schema for Istanbul 3D city models.

Among other issues, large CityGML XML files often cannot be rendered directly on a web browser due to memory constraints. Visualising CityGML over the web requires separating the geometric information from the semantic information in the commonly used 3D graphics formats like KML, COLLADA, glTF and 3D Tiles. It should be noted that when CityGML data is converted to those formats for 3D data visualisation over the Web, often the rich semantics of CityGML are lost (Arroyo et al, 2018). Therefore, in this study a Web services is utilised for attributes queries while 3D Tiles as 3D data visualisation as shown in Figure 6.



3b6bc639-76eb-4d51-9b1f-11ab4f5679b6

objectid	288
ilcekn	1823
mahallekn	40593
kimlikno	0
binakodu	487464547
tip	3
durum	1
zeminustuk	3
zeminaltik	1
yapi_id	118060205
proje_duru	0
yapi_kayna	0
insaat_tur	2
isyeri_say	1
konut_sayi	0
obje_kod	173
otopark	1
globalid	{4863B003-9789-4ED5-9A78-13FB0229D1A1}
ibb_kimlik	118060205
proje_du_1	0
shape_leng	445.882943506
shape_area	7597.93879009

Figure 6. 3D visualization and attributes query of the 3D City Model of Istanbul through Cesium and Web Feature Service (WFS)

The 3D Tiles technique has been proposed as a standard in OGC Standard (OGC, 2016). The technique offers better than glTF in the following aspects – spatial data structure, metadata, and declarative styling for streaming massive heterogeneous 3D geospatial datasets. The technique also supports existing 2D and 2.5D streaming of map tiles and terrain including support for 3D buildings models (exterior and interior), trees, point clouds, TIN terrain, and vector data. Major advantages of 3D Tiles are optimized data streaming and rendering.

3.6..Some Issues and Challenges

Attribute and 3D geometry linkage
Previously, the database shows no linkages between 3D models and attributes. Consequently it produces incomplete information of the area. We overcome this particular issue by having footprint layers into account.

Complex 3D Objects
Most complex 3D objects such as mosques were not properly handled by the existing CityGML exchange format (version 2.0). However this particular issue could be partly managed by introducing tool such as CityGrid (as an intermediate platform). Figure 7 illustrates the visualisation of LoD2 and "LoD2+", whereas Figure 8 shows the separated components of the 3D models.

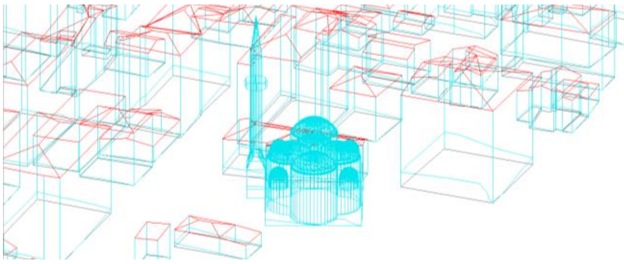
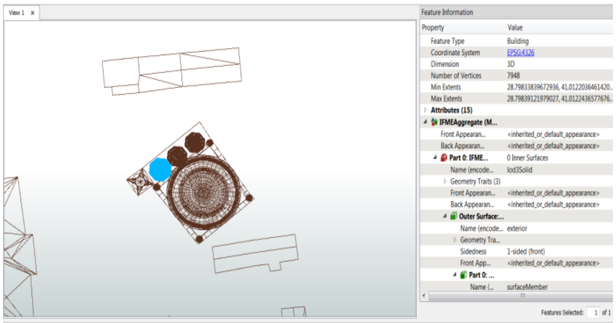


Figure 7. Example of LoD2 buildings and LoD2+ building (mosque)



Feature Information

Property	Value
Feature Type	Building
Coordinate System	EPSG:4326
Dimension	3D
Number of Vertices	7948
Min Extents	28.79833839672936, 41.0122036461420...
Max Extents	28.79839121979027, 41.0122436577676...
Attributes (15)	
IFMEAggregate (M...	
Front Appearan...	<inherited_or_default_appearance>
Back Appearan...	<inherited_or_default_appearance>
Part 0: IFME...	0 Inner Surfaces
Name (encode...	lod3Solid
Geometry Traits (3)	
Front Appearan...	<inherited_or_default_appearance>
Back Appearan...	<inherited_or_default_appearance>
Outer Surface:...	
Name (encode...	exterior
Geometry Tra...	
Sidedness	1-sided (front)
Front App...	<inherited_or_default_appearance>
Part 0: ...	
Name (...	surfaceMember

Features Selected: 1 of 1

Figure 8: Problem with 3D model (mosque) – the structure is divided into separated components instead of one combined object.

4. CONCLUSION

We have developed 3D Database for Istanbul City based on PostgreSQL and 3DCityDB. We also utilised the CityGML exchange format v2.0 for the 3D database schema (mainly for LoD1 and LoD2+) models. 3D objects beyond LoD2+ could not properly handled by the existing CityGML v2.0. We found that 3D Tiles technique for visualisation of city models with Web environment produces better streaming and rendering then glTF and Collada.

We anticipate commercial package like Oracle able to handle more data. Thus combination of Oracle with 3DCityDB could be utilised for much bigger datasets and applications.

REFERENCES

Abdul-Rahman, A. and M. Pilouk (2009). Introduction to 3D GIS. Springer. Heidelberg, Germany.

Arroyo Ohori, K., Biljecki, F., Kumar, K., Ledoux, H., & Stoter, J. (2018). Modeling Cities and Landscapes in 3D with CityGML. 10.1007/978-3-319-92862-3_11.

El Haje, N., Jessel, J.P, Gaildrat, V., & Sanza, C. (2016). 3D Cities Rendering and Visualisation: A Web-Based Solution. In: Eurographics Workshop on Urban Data Modelling and Visualisation (Liege, Belgium).

Gröger G, Kolbe TH, Nagel C, Häfele KH (2012). OGC City Geography Markup Language (CityGML) Encoding Standard, Version 2.0, OGC Doc No. 12–019. Open Geospatial Consortium. https://portal.opengeospatial.org/files/?artifact_id=47842. Accessed 1st April 2019.

Koukofikis, A., Coors, V., & Gutbell, R. (2018). Interoperable Visualization of 3D City Models using OGC's Standard 3D Portrayal Service. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences. IV-4. 113-118. 10.5194/isprs-annals-IV-4-113-2018.

OGC, (2016). OGC is considering a proposed work item for 3D Tiles as a Community Standard, http://www.opengeospatial.org/pressroom/pressreleases/2466. Accessed 1st April 2019.

PostGIS.net, (2019). Spatial and Geographic objects for PostgreSQL, About PostGIS https://postgis.net/

Peyami, B., G. Buyuksalih, A. Abdul Rahman (2019). 3D city modelling of Istanbul – issues, challenges and limitations. GIM Magazine, January.

Santhanavanich, T., Schneider, S., Rodrigues, P., & Coors, V. (2018). Integration and Visualization of Heterogeneous Sensor Data and Geospatial Information. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences. IV-4/W7. 115-122. 10.5194/isprs-annals-IV-4-W7-115-2018.

Takase, Y., N. Sho, A. Sone, and K. Simiya (2011). Automatic generation of 3D city models and related applications. ISPRS Archives, Vol. XXXIV-5/W10, pp.

Yang, Bisheng & Li, Qingquan & Li, D. (2000). Building Model Creating and Storing in 3D Urban GIS.

Yao, Z., C. Nagel, F. Kunde, G. Hudra, P. Willkomm, A. Donaubauer, T. Adolphi and T. H. Kolbe (2018). 3DCityDB – a geodatabase solution for the management, analysis, and visualization of semantic 3D city model based on CityGML. Open Geospatial Data and Standards. 3:5

Revised August 2019

APPENDIX A: Database schema of the Istanbul 3D city models based on CityGML standard