

Hairi Karim¹, Alias Abdul Rahman², Suhaibah Azri³, Zurairah Halim⁴

The Development of Multi-scale Data Management for CityGML-based 3D Buildings

Abstract: The CityGML model is now the norm for smart city or digital twin city development for better planning, management, risk-related modelling and other applications. CityGML comes with five levels of detail (LoD), mainly constructed from point cloud measurements and images of several systems, resulting in a variety of accuracies and detailed models. The LoDs, also known as pre-defined multi-scale models, require large storage-memory-graphic consumption compared to single scale models. Furthermore, these multi-scales have redundancy in geometries, attributes, are costly in terms of time and workload in updating tasks, and are difficult to view in a single viewer. It is essential for data owners to engage with a suitable multi-scale spatial management solution in minimizing the drawbacks of the current implementation. The proper construction, control and management of multi-scale models are needed to encourage and expedite data sharing among data owners, agencies, stakeholders and public users for efficient information retrieval and analyses. This paper discusses the construction of the CityGML model with different LoDs using several datasets. A scale unique ID is introduced to connect all respective LoDs for cross-LoD information queries within a single viewer. The paper also highlights the benefits of intermediate outputs and limitations of the proposed solution, as well as suggestions for the future.

Keywords: CityGML, Level of Details (LoD), multi-scale, scale unique ID, cross-scale query, spatial scale data management


Received: 4 August 2021; accepted: 18 October 2021

© 2022 Authors. This is an open access publication, which can be used, distributed and reproduced in any medium according to the Creative Commons CC-BY 4.0 License.

¹ Universiti Teknologi Malaysia, Faculty of Built Environment and Survey (FABU), 3D GIS Lab, Johor Bahru, Malaysia, email: wmhairigis@gmail.com,  <https://orcid.org/0000-0002-4948-4232>

² Universiti Teknologi Malaysia, Faculty of Built Environment and Survey (FABU), 3D GIS Lab, Johor Bahru, Malaysia, email: alias@utm.my,  <https://orcid.org/0000-0001-5263-8266>

³ Universiti Teknologi Malaysia, Faculty of Built Environment and Survey (FABU), 3D GIS Lab, Johor Bahru, Malaysia, email: suhaibah@utm.my,  <https://orcid.org/0000-0001-7926-9608>

⁴ Cadastral Division, Department of Survey and Mapping Malaysia (JUPEM), Kuala Lumpur, Malaysia, email: nurzurairah@gmail.com,  <https://orcid.org/0000-0003-3650-9485>

1. Introduction

Spatial scale modeling, either in 2D or 3D, aims to create abstraction objects/models in various domains for improved understanding and decision-making in managing the real world. However, abstraction of the model normally comes with various accuracies, needs, and levels of information to be stored. An object such as a building can be represented at many levels of detail and dimensionality, including 0D as Point of Interest (POI), 2D as a building footprint, and 3D as a three-dimensional measurement. The higher the dimension, the more details there are, and the closer it is to the real-world phenomena. For example, in an urban setting, a 3D model will alleviate certain difficulties with photovoltaic (PV) installation on building façades [1]. A building may be represented in CityGML (version 2) as 5-levels of detail, with LoD0 being 2D and the rest being 3D models.

The CityGML is an XML formatted standardized data model introduced by the Open Geospatial Consortium (OGC). This standardized data model enabled exchanging the format of 3D models of city and landscape features from and to various applications and platforms. These exchangeable formats are applicable but not limited to mapping, cadaster, environment, navigation, urban planning, architecture, real estate, simulation, urban facilities management, among other things. The CityGML model (version 2.0) is available in five LoD standards, as illustrated in Figure 1. LoD0 is basically a building footprint with a 2.5D elevation model, while LoD1 is a simple block model without any customized details, and LoD2 is an extension of LoD1 which incorporate roof structures. LoD3 is closer to an architectural model with openings such as windows, doors, façades and rooftops with optional textures. Lastly, LoD4 completes the LoD3 building model by incorporating the interior structures such as rooms, interior doors, stairs, and furnishings.



Fig. 1. Five LoDs introduced in CityGML 2.0

Source: [2]

Handling multiple LoD models in separated file-based architecture is inefficient in terms of storage, attribute information, data updating, and information query, especially if it involves visualization platforms. It also limits the ability of stakeholders, agencies, and public users of their respective domains to share data (e.g. upload and download operation or using Application Programming Interface (API) of an online system). Starting with data collection from various systems, modeling techniques, software and tools with different precision accuracy levels until migration

to a database, many quality controls and integration procedures are required. At present, there is no technique which can incorporate all LoDs of the same building into an enable relational-query which can be accessed by a single map viewer. This paper, therefore, presents an approach to minimize and overcome these limitations. The proposed solution starts with constructing the CityGML model with various LoDs, migration to a database, and retrieval of information cross-LoDs. This paper also introduces a new scale unique ID which acts as a primary key to connect different LoDs and enables cross-LoD queries within the database. The findings of this study is expected to be used as a guideline for implementing new or reorganized existing CityGML multi-scale models at the national level.

The paper is structured as follows: Section 2 highlights the relevant work, followed by Section 3, which describes the datasets used in this study, including the 2D cadaster, Airborne Laser System (ALS), Mobile Laser System (MLS), and building image texture. The method for constructing each CityGML LoD and the scale unique ID is covered in Section 4. Section 5 discusses the outputs of the proposed solution, where they are applied in the 3D cadaster domain as one of Malaysia's national initiatives, known as "SmartKADASTER" [3, 4]. Section 6 concludes the study with recommendations for the future.

2. Related Work

Different users and applications require different abstractions of the spatial object [5], creating multiple layers of datasets and models to be managed, shared, and maintained. Current solutions at the level of spatial abstraction have utilized multi-scale (discrete approach), vario-scale (continues) and generalization algorithms to serve users with information level in a map viewer. However, only a multi-scale solution provides specific data sharing through different user's domains, cross-operation requirement details, and multiple formats. It is designed for data custodians and owners to share their datasets with various levels of accuracies, formats, and details as required by stakeholders, clients and public users; especially for 3D buildings. Thus, a standard development guideline is required for each pre-defined 3D building LoD to encourage users in various applications such as in urban planning, mapping, sustainable development and the smart city. Besides that, most domains have their customized LoD definitions and specifications to support their operations or analyses, demanding different abstractions and models of real-world objects [6]. The current methods for handling multi-scale data require a file-based proprietary format and generalization for either 2D and 3D vector datatypes.

Several studies on the CityGML multi-scale have been carried out [7–9]. For example, [10] models the ancient city of Taranto in Italy with multi-scale LODs to simulate the urban changes from the mid-1800s until the present. Graphic and iconographic documents have been utilized to construct the LOD1 model, which

focuses on building blocks, while the LOD3 model is focused on street furniture such as bridges and flyovers. The LOD3 is constructed based on a set of point clouds acquired using an airborne laser system (ALS). Although the LOD1 model is constructed to visualize the building blocks in different time frames, the series needs to be inspected individually in a different mapviewer. Their study, however, did not address the dependencies between geometric entities of different LODs. CityGML standards state that each user is responsible for ensuring the modeled objects (in different LODs) refer to the same objects in the real world [11]. This indicates that the consistency of objects with different LODs needs to be validated. The level-of-details for the objects must be interrelated in terms of data updating and modification. Consequently, it requires an approach or procedure to produce a multi-scale model that is appropriate on coarser or finer LODs.

The LoDs multi-scale data and models are normally constructed via large area and precise measuring systems during data collection (e.g. point cloud). It also involves post-processing before constructing each LoD 3D model using the requisite software. Normal practices utilize LiDAR 360, AutoCAD Revit, Blender and FME Safe software to construct the 3D LoD models according to the user's operation requirements. For example, for spatial mapping, [12] and [13] utilize FME software to convert multiple sources of datasets to construct 3D CityGML LoD models. [14] also integrates various data sources in FME to produce the CityGML model for urban flood management. Since there are various data collection methods available with different accuracy levels, software (and integration between software/formats), workflows, and user specifications; a multi-scale standard should be applied such as in CityGML to enable sharing among other potential users, thus minimizing the cost of duplicated data collection for each application/domain). The sharing platform could be an online map service (paid/free depending on requested LoD details) or a custom developed system or through database sharing of API integration. Currently, most multi-scale datasets/models such as CityGML are in a file-based format, making then inaccessible to others with secure authentication control.

In Malaysia, the Department of Survey and Mapping Malaysia (JUPEM) initiated a pilot study of implementing a multipurpose cadaster in 2012. The pilot study advanced to phase 2 in 2020 with the addition of a mapping and geospatial information system based on the CityGML 2.0 schema. The value-added application is designed to support Smart City implementation in Malaysia, which offers the ability to spatially identify assets and to represent city dynamics in 3D, as well as accurate cadaster information. The development of the smart decision-making system will support the management of land in a more effective and efficient way [15]. A rich data model benefits the user due to the high demand for structured storage, management, and analysis of data. Nonetheless, the previous SmartKADASTER system (Phase 1) has limitation in terms of the system's capability to support different LoDs and users, due to the storage of 3D city models in the form of a file-based system.

Realizing these setbacks, phase 2 of SmartKADASTER aims to ensure that the improved system complies with the city model standard and encapsulates the 3D city model into the database based on the reviews from previous researchers [16–21], targeting beyond the cadaster purpose and format interoperability. This paper suggests how to improve the management of information using a database solution since it is better, more practical, accessible, and more coherent than file-based systems [21].

3. Datasets

This work utilizes a square grid of 6.25 km² (2.5 km × 2.5 km area) testing datasets in Klang, Selangor. Primary datasets include cadaster 2D lots, ortho-photos, LiDAR from ALS and MLS (with side camera for texture). All these data are obtained from JUPEM, one of the 230 grids in the SmartKadaster project (Phase 2). To process and create the model from these datasets, several pieces of software and tools are used, namely LiDAR 360, Google SketchUp (with a commercial extension of Undet and S4U Slice), Adobe Photoshop (for texture editing), FME workbench, 3D CityDB tool, and PostgreSQL database, which will be discussed in section 4.

3.1. Airborne Laser System (ALS)

The ALS system is used to capture point clouds, oblique and nadir aerial photos which later produces orthophoto (Fig. 2) and other products (Fig. 3). Ground Control Points (GCPs) and some cadaster boundary marks were used for georeferencing the point clouds to the ground survey.



Fig. 2. The generated orthophoto of the study area (10-cm resolution)

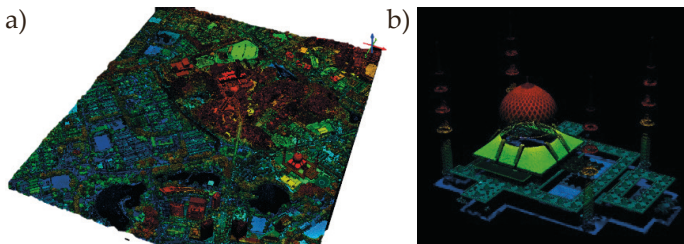


Fig. 3. Example of ALS point clouds: a) the study area; b) a selected building extracted point clouds (cropped) for 3D modelling

3.2. Mobile Laser System (MLS)

The MLS system is used to capture street-based point clouds of building façades as well as the image texture and image of the buildings. Buildings covered by more than three sides of the façade (accessible by road) are selected for modelling in CityGML LoD2 and LoD3. The maximum height measurement of MLS point clouds can reach up to 5 floors (20 m), highly dependent on the offset distance between an access road and the measured building. Figure 4 shows point clouds from the MLS system, while Figure 5 shows some examples of photos from the MLS side camera for the building texture.

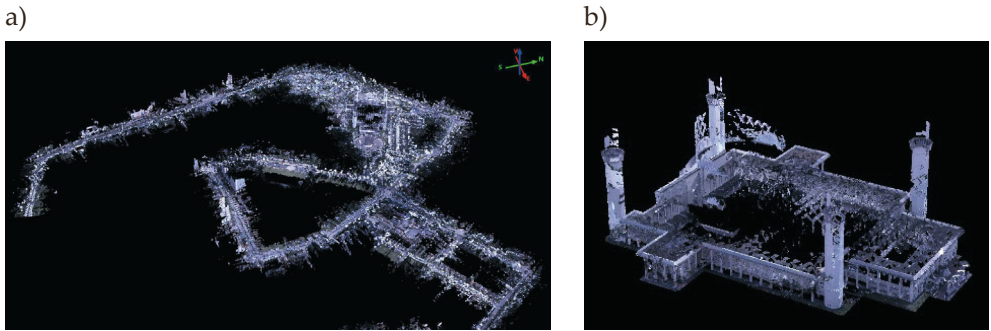


Fig. 4. Example of a track of MLS mission: a) 3 km length with point cloud; b) the same building point clouds is extracted (cropped) from the MLS



Fig. 5. Examples of building textures from side camera of MLS system

3.3. Combining ALS and MLS Datasets

Using LiDAR 360 software, the cropped point clouds data (.las) of a specific building from ALS and MLS were merged into a single file. A quality check was carried out to see if the combination of ALS and MLS is accurately matched or if it falls outside the model's acceptable tolerance. Figure 6 shows the relevant datasets as well as the merged point cloud's results. The newly generated file is then

ready for the 3D construction modelling process in Google SketchUp using the Undet extension tool (a commercial extension used to load and view point clouds in SketchUp).

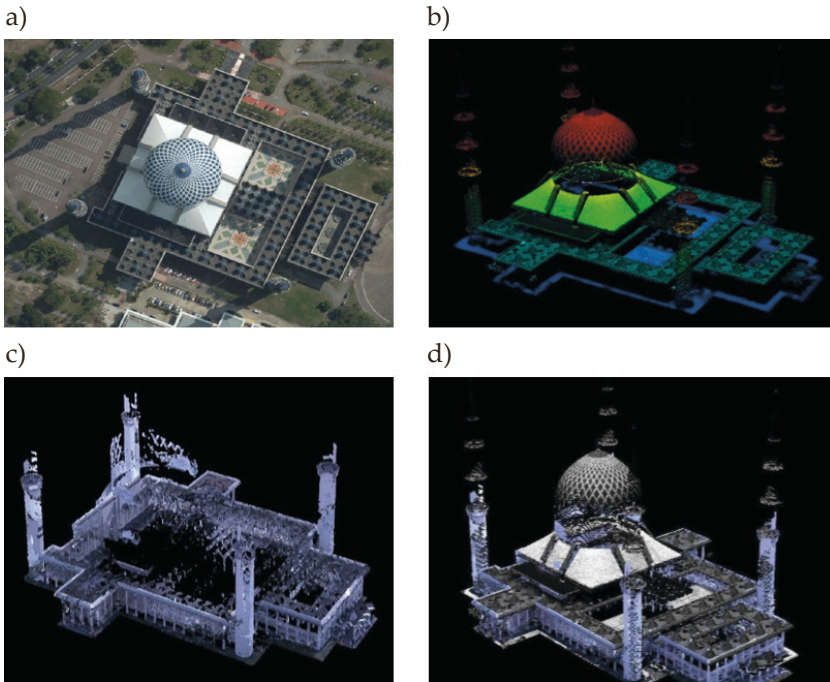


Fig. 6. Examples of cropped datasets for a building (Sultan Salahuddin Abdul Aziz Mosque) respectively: a) orthophoto; b) ALS point clouds; c) MLS point clouds; d) ALS-MLS colored point clouds combination

4. Method

The overall work is divided into several processes; preliminary model preparation, setting upscale, unique ID environment and cross-scale LoD query. We introduced a term called “Scale Unique ID” to denote the construction of building model in several LoDs which comply with the CityGML schema and effective data management solutions to minimize current implementation drawbacks. It is crucial to create and maintain the relationship between multiple LoD representations with the corresponding real object. The scale unique ID enables cross-scale queries (LoDs) especially to support single map viewer. The workflow consists of four main phases:

1. data source,
2. CityGML construction with scale unique ID,
3. database query,
4. single visualization.

The first phase (data source) was covered in Section 3, while phase two is addressed in Section 4.1 (CityGML Development). The implementation of the scale unique ID (shown in red in Figure 7) is described in Section 4.2.

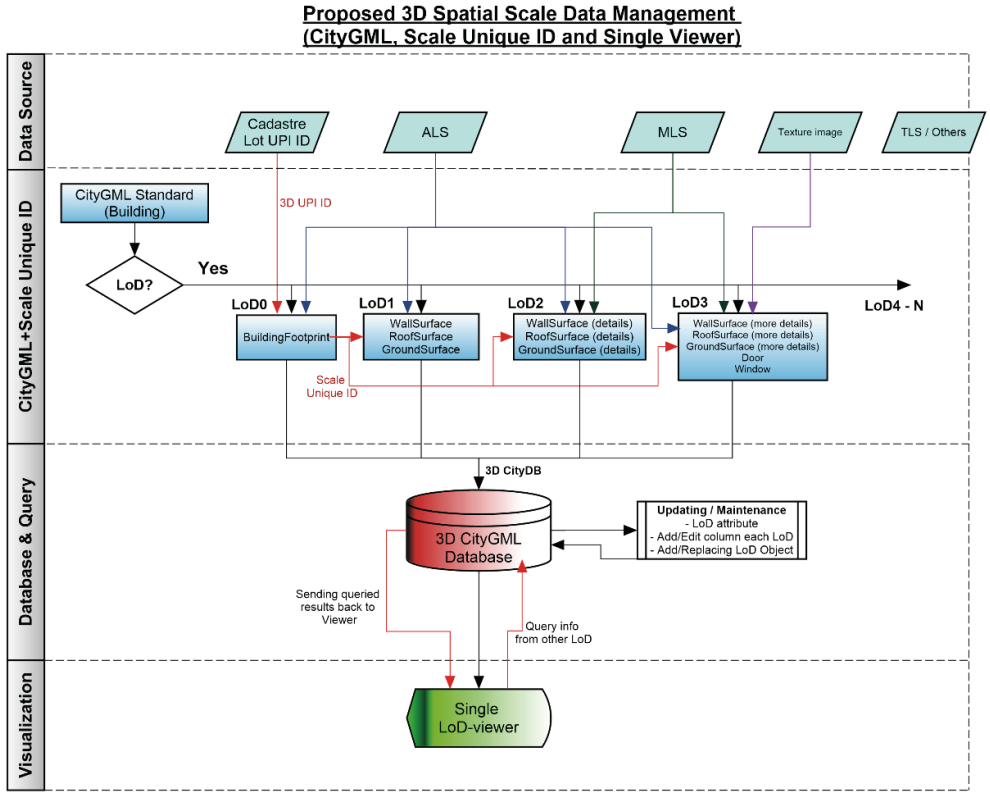


Fig. 7. The overall workflow for 3D spatial multi-scale data management

The scale unique ID is a primary key to support cross-scale information queries (e.g. from other LoD attribute tables). It is essential to prepare a model with relational connection for the next phase of developing a single viewer for users to access information. The fourth phase of this workflow will not be discussed in depth because the integrated LoD visualization is still not available. There are currently no visualization platforms to directly access CityGML models within a database (e.g. PostgreSQL), either as a Web application (e.g. Cesium 3D) or desktop-based software (e.g. FME).

4.1. Development of Multi-scale CityGML Models

The construction of LoD0, 1, 2, 3, and 4 models is not identical to each other, since each model uses different input datasets, software, and techniques. Based on the process workflow and complexity of each model, the construction of models can

be grouped into three categories: LoD0 and LoD1, LoD2 and LoD3, and LoD4. The latter is excluded in this section, due to unavailability of the dataset. Figure 8 shows an example of LoD2 model construction while Figure 9 illustrates a construction for LoD3 using Google SketchUp software as part of completing five LoDs of CityGML standards as shown in Figure 10.

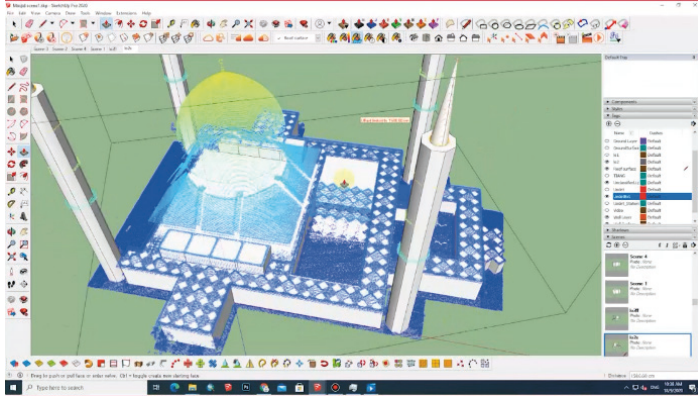


Fig. 8. Example of a LoD2 model construction (rooftop geometry and texture) using SketchUp software

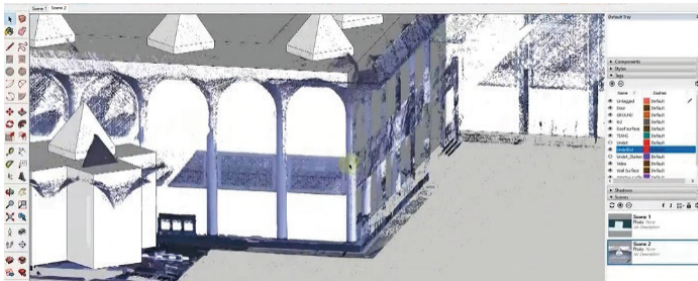


Fig. 9. Example of a LoD3 model construction (façade texture and building openings)

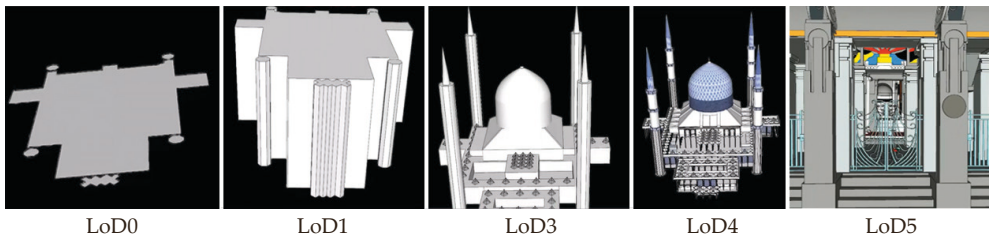


Fig. 10. Five LoDs in the CityGML 2.0 standard, implemented for the building of Sultan Salahuddin Abdul Aziz Mosque

Source: [22]

Combining ALS and MLS Datasets

CityGML LoD1 is a solid-based 3D model generated automatically from the building footprint LoD0 (Fig. 11, the building outlined in green with a manual digitizing process) using FME workbench software. The LoD0 of the study area and the ALS point cloud (height references for extrusion as per building block) are used as the input data. The process of model generation is illustrated in Figure 12. The LoD1 buildings are shown in Figure 13. The height of each building block varies depending on the mean of the point cloud elevation values that fall within a building footprint.

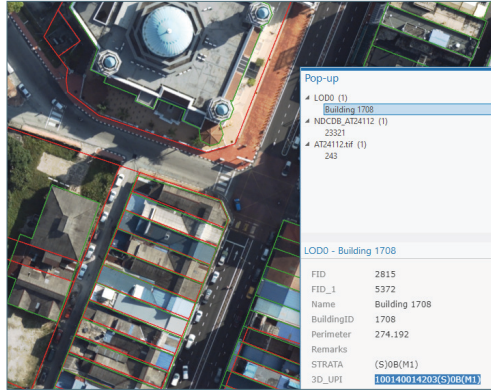


Fig. 11. Example of LoD0 (building footprint, light green vector) in relation to the 2D lot (colored red)

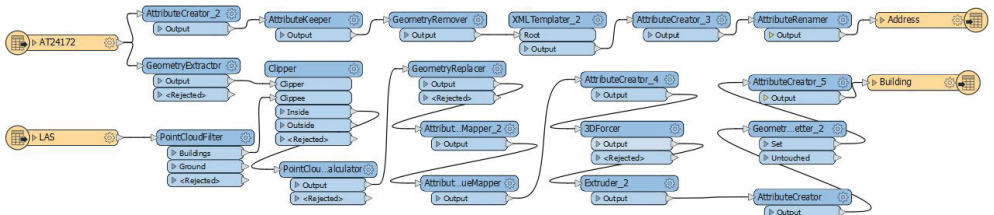


Fig. 12. Model builder of automatic LoD1 generation from LoD0 and ALS in FME workbench

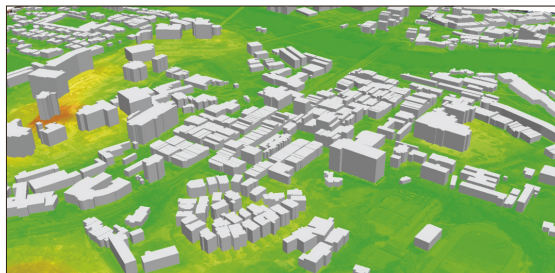


Fig. 13. The generated LoD1 buildings (draped with orthophoto and DEM)

LoD2 and LoD3

The LoD2 and LoD3 models are constructed using different process workflows and software to the LoD1 processes. The construction mostly requires manual measuring, editing, and quality checking in Google SketchUp software. The Undet tool is used to plot the combined ALS (rooftop), and MLS (façade) point clouds with the actual coordinate system for the modelling process in SketchUp. The overall model construction processes are illustrated in Figure 14, while workflow for generating texture is specifically for LoD3.

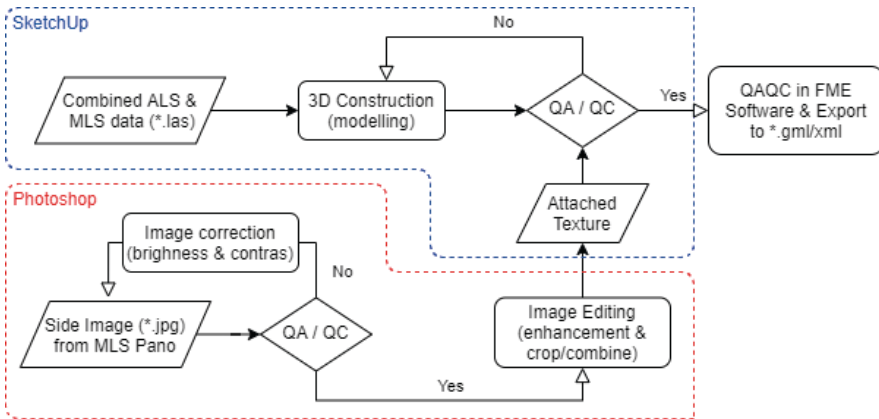


Fig. 14. General workflow for LoD2 and LoD3 building model construction

An example of the LoD2 model is shown in Figure 15, which features a building in the study area. Figure 16 shows an example of the constructed LoD3 buildings (with and without texture) before the migration process to CityGML schema using FME software.

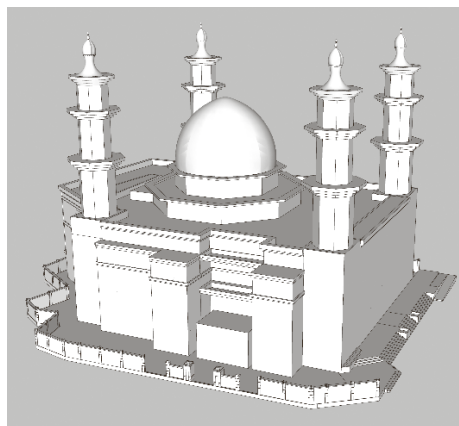


Fig. 15. Model of a mosque in LoD2

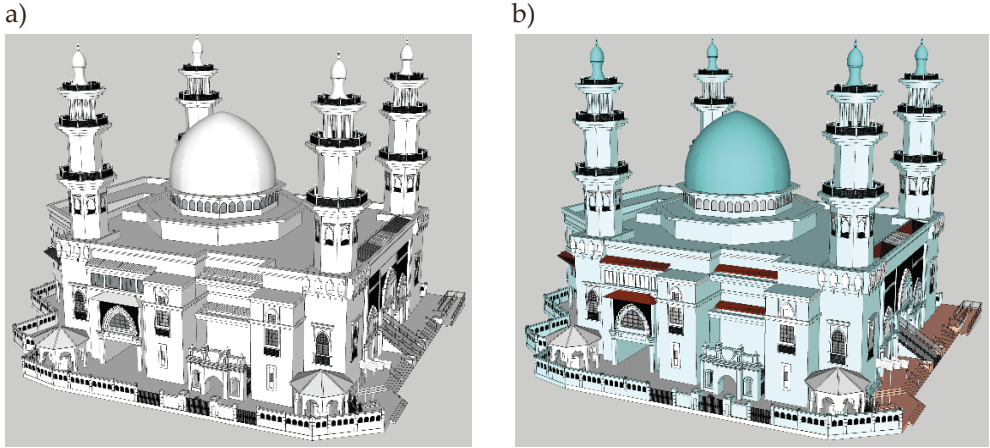


Fig. 16. Model of a mosque in LoD3: a) without textures; b) with textures

4.2. Development of Multi-scale CityGML Models

A new ID, called a scale unique ID is introduced in this workflow to connect each of the respective LoDs back to a single basic 3D unit or, in other words, to reconnect them as a building from multiple building representations. The ID meets object-oriented database (OODB) requirements and is well compatible with the City-Object database schema for CityGML. The ID is an extended version of the existing Unique Parcel Identification (UPI) for the 2D cadaster lots. The current 2D UPI ID for Malaysia’s cadaster lot is shown in Figure 17. It combines the state code, district, sub-district, section, and lot number into a 2D parcel.

OBJEC	Shape *	NEGERI	DAERAH	MUKIM	SEKSYEN	LOT	UPI	KELUASAN	KOD_KEGUNAANTANAH	PA	STATUS	NOFAILUKUR	NAMAPEMLIK
1	Polygon	10	01	75	000	84012	10017500084012	308.598	<Null>	PA96805	65	PUBLSEL1531_2003	<Null>
2	Polygon	10	01	75	000	81533	10017500081533	130.062	2	PA72242	65	<Null>	<Null>
3	Polygon	10	01	75	000	84030	10017500084030	306.6	<Null>	PA96806	65	<Null>	<Null>
4	Polygon	10	01	75	000	84032	10017500084032	411.283	<Null>	PA96806	65	<Null>	<Null>
5	Polygon	10	01	75	000	84033	10017500084033	308.59	<Null>	PA96806	65	<Null>	<Null>
6	Polygon	10	01	75	000	83206	10017500083206	9589.262	<Null>	PA72269	65	<Null>	<Null>
7	Polygon	10	01	75	000	83162	10017500083162	308.601	<Null>	PA72267	65	<Null>	<Null>
8	Polygon	10	01	75	000	81482	10017500081482	143.08	2	PA72244	65	PUBLSEL276_1999	<Null>
9	Polygon	10	01	75	000	81485	10017500081485	143.08	2	PA72244	65	PUBLSEL276_1999	<Null>
10	Polygon	10	01	75	000	81488	10017500081488	353.503	2	PA72244	65	PUBLSEL276_1999	<Null>
11	Polygon	10	01	75	000	81492	10017500081492	143.08	2	PA72244	65	PUBLSEL276_1999	<Null>
12	Polygon	10	01	75	000	83160	10017500083160	306.601	<Null>	PA72265	65	<Null>	<Null>
13	Polygon	10	01	75	000	81443	10017500081443	143.08	<Null>	PA72244	65	<Null>	<Null>

Fig. 17. Example of cadaster lots layers and the UPI ID for respective lots

As illustrated in Figure 11, LoD0 (building footprint) is a subset of 2D cadaster lot, where the 2D UPI ID could be further extended toward a more detailed subset ID (e.g. 2D building footprint in LoD0). However, to embark on 3D buildings,

some scenarios should be considered, such as in Figure 18, in which a 2D UPI ID is evolved to become a 3D UPI ID. At present, JUPEM has already implemented 3D UPI ID for 3D buildings as unique ID – especially for cadaster strata ownership. Thus, the 3D UPI ID (single building) is utilized as the basic core ID to be extended toward a scale unique ID. Table 1 indicates scenarios of the existing 3D UPI ID and the proposed scale unique ID.

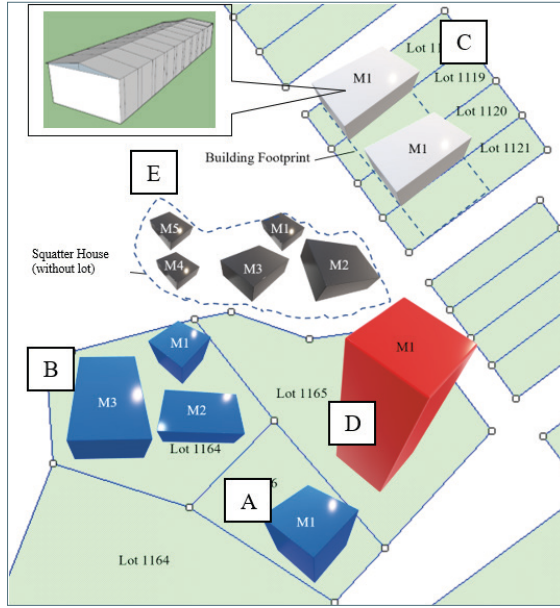


Fig. 18. Example of scenarios in 3D cadaster lot

Table 1. Example of scenarios, 3D UPI ID and the proposed scale unique ID; where (S)_ is strata information, and B(M_) is block/building with model or tower. D0 represents the scale unique ID extension for LoD in CityGML

Scenario	Existing 3D UPI ID	Scale unique ID (LoD0)
Single building within a cadaster lot (scenario A)	10017500081443(S)0B(M1)	10017500081443(S)0B(M1)D0
Multi-buildings (scenario B) and Shared wall within the single lot (scenario C)	10017500081443(S)0B(M1) 10017500081443(S)0B(M2) 10017500081443(S)0B(M3)	10017500081443(S)0B(M1)D0 10017500081443(S)0B(M2)D0 10017500081443(S)0B(M3)D0
Multi-level building (strata) in a lot (scenario D)	10017500081443(S)1B(M1) 1B(M2) if with 2 towers	10017500081443(S)1B(M1)D0
Buildings with no cadaster lot	10017500081443 TM00001(S)0B(M1)	10017500081443 TM00001(S)0B(M1)D0

For LoD1, the scale unique ID of LoD0 (D0) will be replaced with LoD1 (D1) during the extrusion in FME (previously shown in Figure 11) or in the database table (after migration). The general process of model migration from an XML format to PostgreSQL database is illustrated in Figure 19. In the migration process of the building module (CityGML), very minimum attributes of the 3D model from SketchUp could be carried over to CityGML, such as text-based without special characters, and in this case, the scale unique ID. Once the migration is completed, CityObject generates the building Table (dotted line in red), which carries the inserted ID as shown in Figure 20. As a subset of the building table, boxes (outlined in red, green, black, and blue) indicate data storage in the database for 3D building with respective LoDs (0, 1, 2, 3 and 4). The second group of boxes below the database interface shows the terrain intersection with respective LoD 3D building (LoD2 and LoD4 are not available).

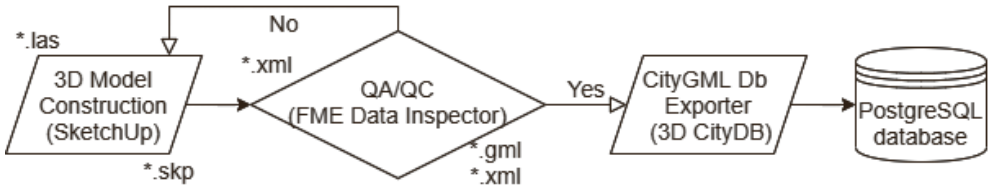


Fig. 19. The workflow of 3D model migration from model construction to PostgreSQL

The screenshot shows the CityObject interface with a tree view on the left and a query editor on the right. The tree view shows a 'building' class with 39 columns and 15 constraints. The constraints are listed as follows:

- building_cityobject_fk
- building_lod0footprint_fk
- building_lod0roofprint_fk
- building_lod1msrf_fk
- building_lod1solid_fk
- building_lod2msrf_fk
- building_lod2solid_fk
- building_lod3msrf_fk
- building_lod3solid_fk
- building_lod4msrf_fk
- building_lod4solid_fk
- building_objectclass_fk
- building_parent_fk
- building_pk
- building_root_fk

The query editor shows the following SQL query:

```
1 SELECT * FROM citydb.building
2 ORDER BY id ASC LIMIT 100
3
```

The data output table shows the following columns and rows:

on	lod2_terrain_intersection geometry	lod3_terrain_intersection geometry	lod4_terrain_intersection geometry
	01050000A0E610000001000000...	01050000A0E610000001000000...	01050000A0E610000001000000...
	LoD2	LoD3	LoD4
	01050000A0E610000001000000...	01050000A0E610000001000000...	01050000A0E610000001000000...
	01050000A0E610000001000000...	01050000A0E610000001000000...	01050000A0E610000001000000...

Fig. 20. The structure of LoD in building a thematic class of CityGML in CityObject

The models in LoD1, LoD2 and LoD3 or LoD4 (modeled on different platforms) are migrated (having passed the QAQC phase in FME) using the 3D CityDB tool from an XML/GML format file to the PostgreSQL database. Figure 21 shows the outputs

imported from LoD1 in bulk migration using 3D CityDB, while Figures 21–24 show individual building block imported results for LoD2, LoD3, and LoD4, respectively, in the Postgres database.

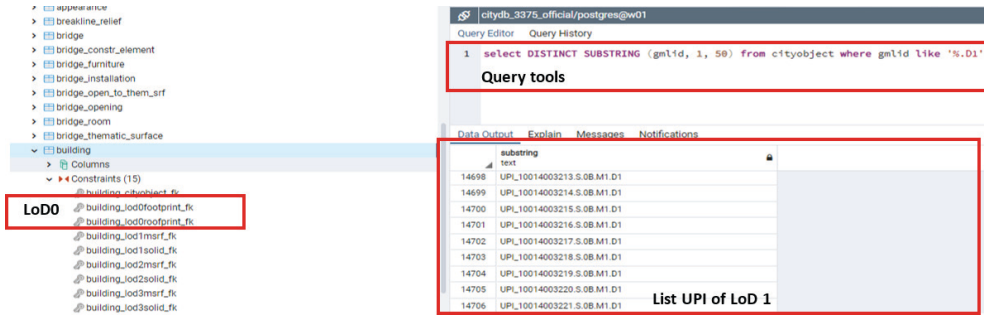


Fig. 21. Scale unique ID for LoD1 after migration

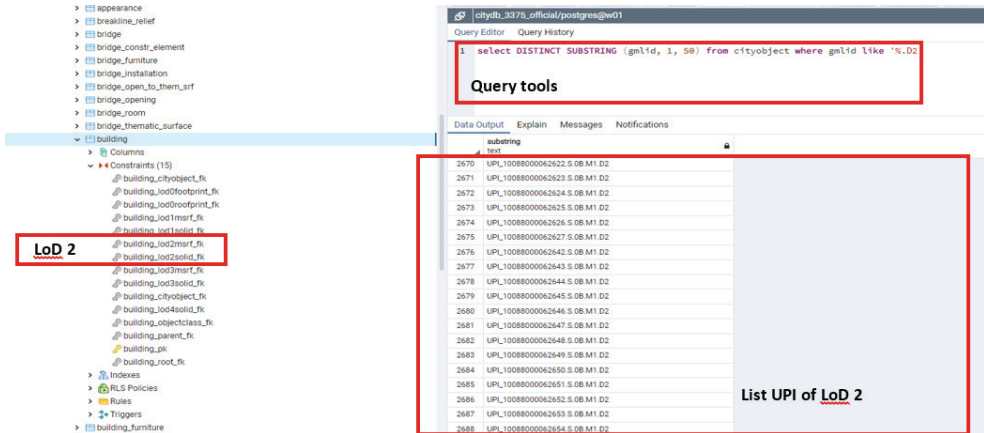


Fig. 22. Scale unique ID for LoD2 after migration

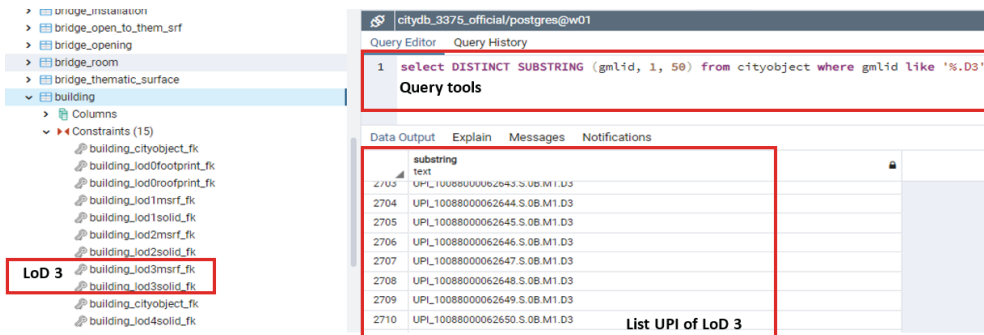


Fig. 23. Scale unique ID for LoD3 after migration

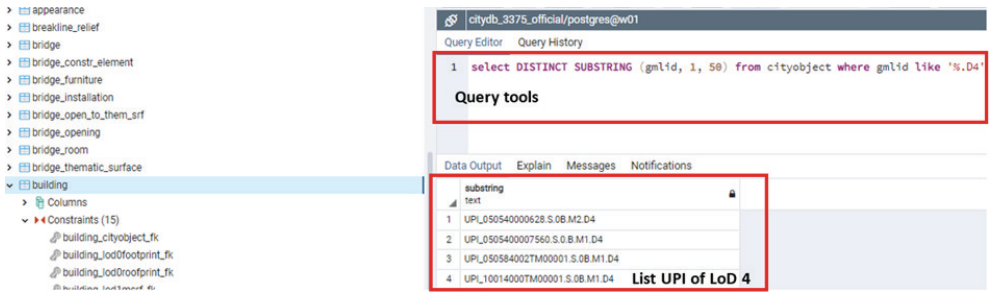


Fig. 24. Scale unique ID for LoD4 after migration

5. Results and Discussion

This section discussed the final results of the CityGML LoD models migration into the PostgreSQL database as well as the capability to execute advanced spatial and attribute queries across the LoD model. Several intermediate models are also considered as secondary outputs of this study, which may be useful in other domains: e.g. the SketchUp file of LoD2 and LoD3 for urban and landscape planners in their respective software applications. The section also includes a discussion on the advantages and potential integration with other datasets for future usage.

5.1. Final Results (Cross-scale Query)

After all of the models have been successfully migrated into the databases as shown in Figures 21–24, the new attribute column can easily be added, modified, or deleted using SQL syntax in the Query tool section. It also allows queries such as selecting a 3D building based on a certain 3D UPI (e.g. 10088000062656.S.0B.M1). The proposed multi-scale spatial data management via scale unique ID (in a database termed gmlid) also enables cross-scale (LoD) information extraction and creates an object-oriented connection with multiple representations and class components of a particular building (Fig. 25).

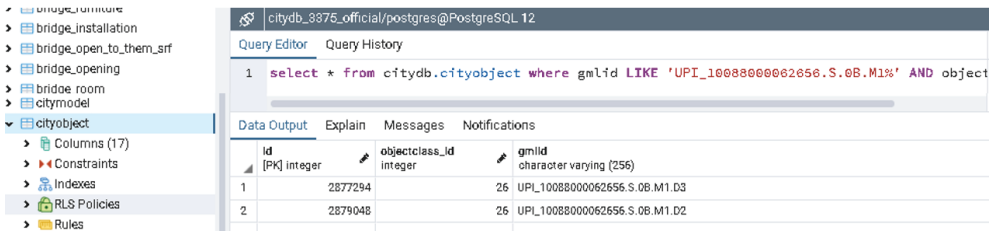


Fig. 25. A building of UPI_10088000062656.S.0B.M1 has two representation LoD2 (...D2) and LoD3 (...D3) as shown in the output query

An example of the query is shown in Figure 25 with:

```
select *
from citydb.cityobject
where gmlid LIKE 'UPI_10088000062656.S.0B.M1%' AND objectclass_ID = 26
```

Since we were using the OODB structure as an object class, we were also able to do detailed information queries. More information on the object information could easily be retrieved in relation to the respective LoDs, including information on the related object sub-class of a selected building component in the respective LoD, such as the number of windows and doors associated with a particular 3D building. The answers are as shown in Figures 26 and 27, respectively. Although only the LoD2 is viewed on the visualization platform, further details on the modelled building such as LoD4 (e.g. room name, size, number of chairs, tables, and computers) could be extracted efficiently. As a result, it significantly reduces the amount of graphics and memory used by the machine while allowing for faster rendering.

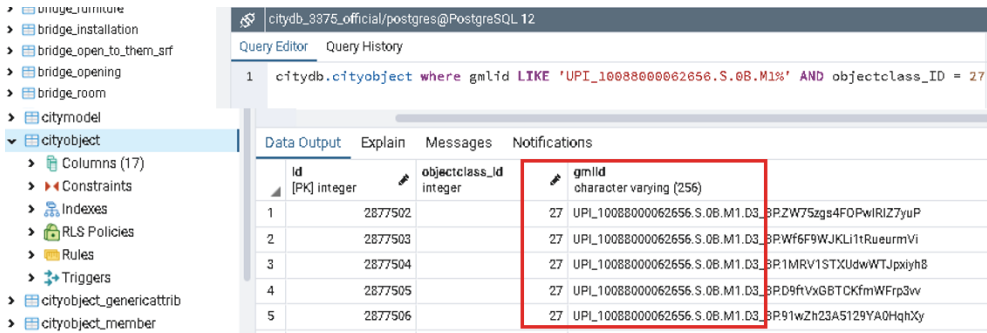


Fig. 26. Example of building UPI_10088000062656.S.0B.M1.D3 with 5 doors (object class 27) in LoD3 representation, but zero in LoD2

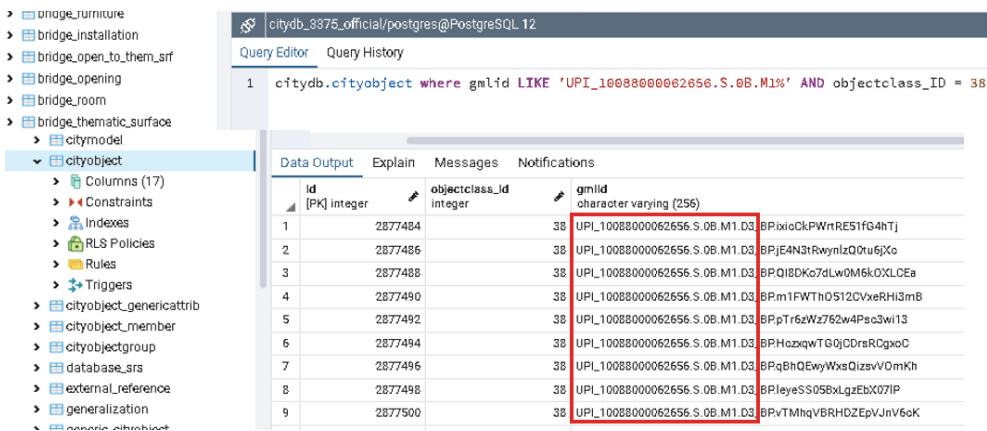


Fig. 27. Example of building UPI_10088000062656.S.0B.M1.D3 with a list of windows (object class 38), but zero in LoD2 (D2)

This cross-scale query capability, which allows users to access information or attributes from each pre-defined LoDs, is a key element in reducing the current multi-scale data management constraint by using a scale unique ID. Its purpose is to ensure data readiness in order to support a single viewer with multiple representation details. The fourth and last phase of the proposed workflow (Fig. 7) for single visualization, however, encounters some limitations with the existing visualization platforms as there is no desktop or web-based application support for this kind of scale-management. For example, the well-known online platform 3D Cesium cannot directly access the 3D PostgreSQL database, thus the model needs to be converted into the 3D tiles format which is file-based. This file-based format is unable to support the scale unique ID or perform a cross-scale query for sub-class information retrieval.

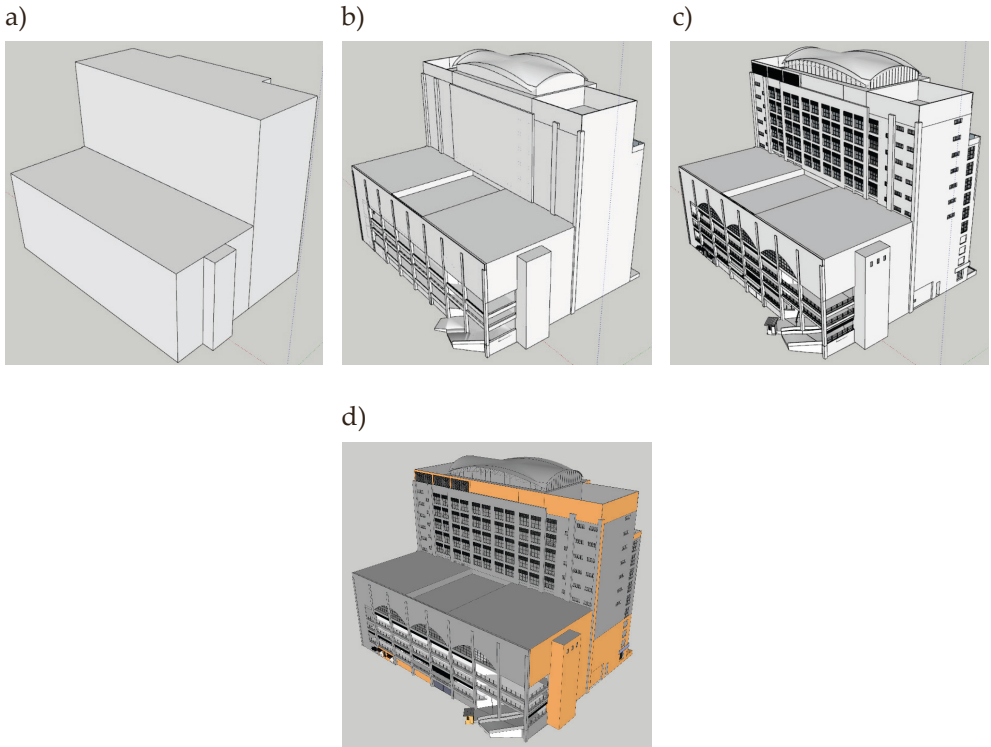


Fig. 28. Example of final models: LoD1 (a), LoD2 (b), LoD3 (c) and LoD3 (d) with textures for a building name of Wisma Perbadanan Kemajuan Pertanian Selangor and are stored in the PostgreSQL database

Aside from the final results in the PostgreSQL database (LoD models such as in Figure 28), several intermediate results (models) and formats were also generated during the process.

5.2. Intermediate Models

The intermediate models refer to the current practice in the industry, which use and maintain the LoD models in respective proprietary formats and software. The intermediate output include the quality control on accuracy and specification of each CityGML LoD standards such as SketchUp files (as in Fig. 29), XML files – FME software, CityGML (*.gml) format and migration results (e.g. Fig. 30). The quality control for each software and tool allows us to minimize errors (caused by the modeler or by software interchange format) and accuracy acceptance tolerance for each LoD as specially requested by clients, specific application needs, national standard, and CityGML international schema. For example, JUPEM only tolerates ±0.3 meter difference between the building model and the measured point clouds for LoD2 and LoD3 to fit their project objective.

The intermediate models also facilitate cross-scale queries of the textured based model in LoD3 and LoD4. For example, in Figure 30, each rooftop texture, façade, and building fixtures such as signboards, are stored and belong to which building scale unique ID within the database. Figure 29 shows the process of model quality inspection in FME software prior to migration. The migration of each CityGML LoD into the PostgreSQL database using 3D CityDB must also be considered, particularly in the case of migration errors and missing objects or mismatch between number of objects before and after migration. Figure 31 shows a log report (3D CityDB), generated by a successful migration.

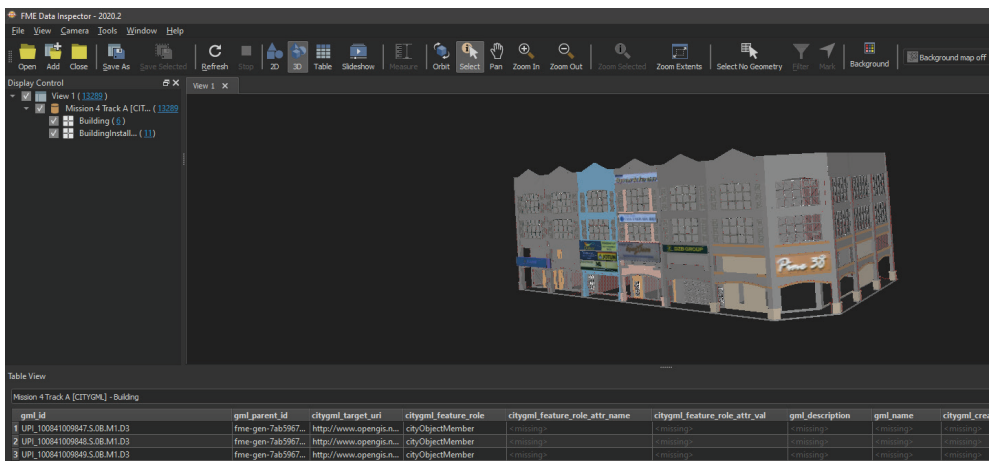


Fig. 29. Example of LoD3 with texture viewed in FME Data Inspector as quality inspection (QC) before import to 3DCityDB

The procedures which integrate LiDAR 360-SketchUp-FME-3D CityDB-PostgreSQL are highly cost-effective and create multiple levels of quality assurance (QA) and, quality control (QC) at the same time. Using this approach, users can easily

monitor and control errors such as missing windows, textures in LoD3 and so forth that arise during model construction and migration from one software to another. The QA and QC of model quality monitoring in stages are essential since the construction of the LoD models involves staffing and heavy workloads as it needs to be done manually especially for LoD3 and, if applicable, for LoD.

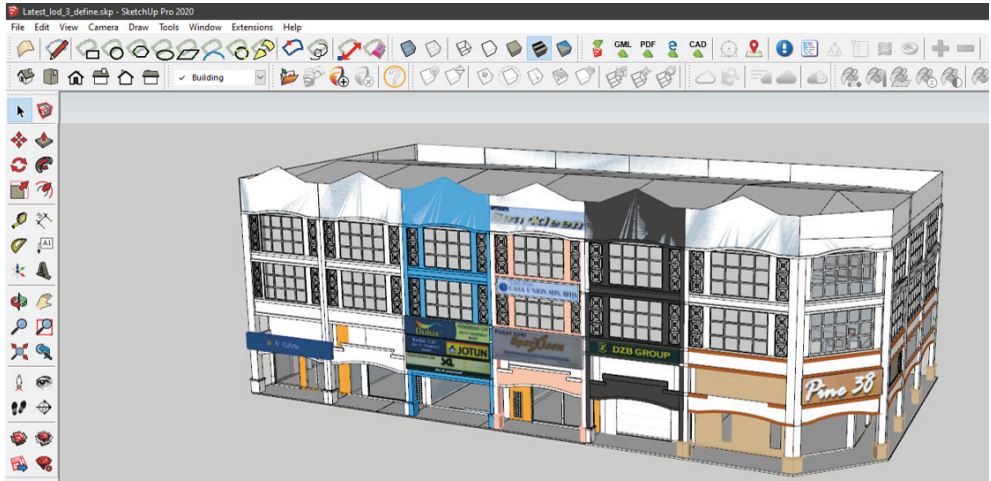


Fig. 30. Example of LoD3, with the texture of a building block separated by building ownership (each lot) as a final result at SketchUp software

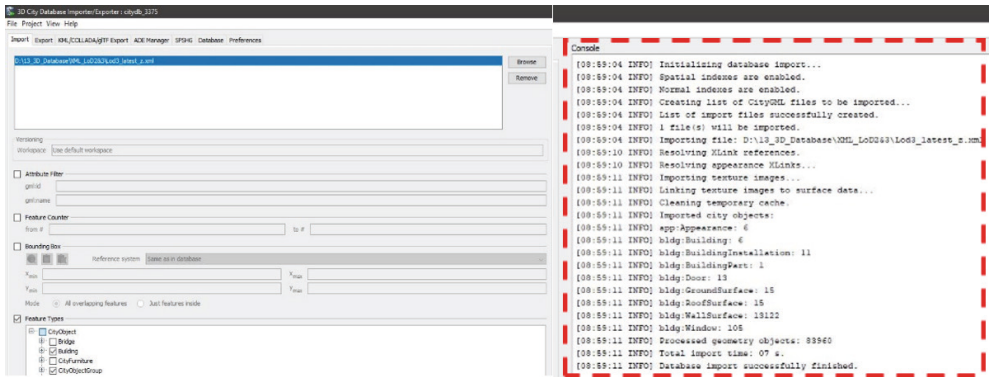


Fig. 31. The LoD3 with texture model successfully imported into 3D database (red dotted box is imported log report including number of building component such as roof, door, windows, ground surface, building installation and others)

The proposed solution makes it easier to maintain and update work on a new LoD building model's geometry and attributes (delete, edit, add) than file-based or generalisation techniques, because users do not have control over updating while the function runs automatically. This is because the scale-unique ID enables faster

searches for the intended LoDs of a particular building and reduces duplicated information key-in for all LoD tables. For multi-scale datasets, the maintenance effort will considerably minimize costs, particularly in terms of time, workload (separate updating for each LoD), and personnel. Reduced redundancy of multiple files allows easier information retrieval regardless of scale. The cost of storage was simultaneously reduced with the reduction of redundancy in storage. Since the implementation of the model utilizes the open-source PostgreSQL database as its core-based architecture, both 2D spatial (using the PostGIS extension) and 3D CityGML (CityObject schema) can be integrated into one visualization platform. The capability of the database in terms of query, level of user's authentication, data security, backup, performance and others are both the same for 2D and 3D, regardless of differences in storage and machine specification.

Intermediate results of this process also produced other 3D file-based formats such as SKP (SketchUp), GML, and XML, which are useful and major input formats for current implementation in a variety of disciplines, such as mapping, urban planning, taxation, and local authority). Models developed with these formats can be used before applying the proposed scale unique ID, and multi-scale data management, if it decided to be implemented in any organization later.

If the new model is constructed using the CityGML 3.0, the new CityGML 3.0 can likewise be integrated later. [23] mentioned that their implementation of the CityGML 3.0 model uses the 3D CityDB (new version) and thus could be in the same CityObject schema of the PostgreSQL database as the current 2.0 model. The scale unique ID should be introduced to each of the newest model for integration (existing 2.0 with new 3.0) and efficient management. For integration (existing 2.0 with incoming CityGML v3.0) and efficient maintenance, the scale's unique ID should be added to each of the newest models. The lack of a visualization platform capable of integrating 3D CityObject schema and data direct access from a database is one of the key limitations of this study. This is because the visualization of a single 3D model capable of accessing information from other LoDs is relatively new within the geospatial research community, since most practices utilize the tiling method and lack 3D database and multi-scale integration. However, we firmly believe that a single layer visualization supporting cross-scale information extraction via live database integration (which is a better solution) will be realized in the near future.

6. Conclusion

This paper describes the development of the LoDs 3D buildings of the CityGML model for better multi-scale data management. It introduces the scale unique ID as one of the techniques to enable cross-scale information query where it can be accessed via a single viewer, such as in a customized 3D Cesium Ion. Updating attributes for each LoDs becomes simpler, manageable, and cost-effective since it reduces storage and maintenance. This work also offers integration of existing multi-scale

2D datasets and/or with 3D models of CityGML 3.0 for the improved sharing of information such as single database and single viewer. New updates and additional LoD models such as LoD4, will be delivered and shared with various users in real-time without the requirement for any versioning. We intend to extend the work primarily on the visualization platform as the current platform can hardly support a single LoD model with cross-scale information query.

Acknowledgement

The authors would like to thank the Department of Surveying and Mapping Malaysia (JUPEM) for the 3D datasets.

References

- [1] Strzalka A., Alam N., Duminil E., Coors V., Eicker U.: *Large scale integration of photovoltaics in cities*. Applied Energy, vol. 93, 2012, pp. 413–421. <https://doi.org/10.1016/j.apenergy.2011.12.033>.
- [2] Biljecki F., Ledoux H., Stoter J.: *An improved LOD specification for 3D building models*. Computers Environment and Urban Systems, vol. 59, 2016, pp. 25–37. <https://doi.org/10.1016/j.compenvurbsys.2016.04.005>.
- [3] Halim A.N.Z, Lim C.K., Karim H.: *Developing a 3D City Model Database Beyond Cadastral Purposes*. GGT 2021, Shah Alam, Malaysia. IOP Conference Series Earth and Environmental Science, vol. 767(1), 2021, 012027. <https://doi.org/10.1088/1755-1315/767/1/012027>.
- [4] Bin Isa M.N., Hua T.C., Binti Abdul Halim N.Z.: *Smartkadaster: Observing Beyond Traditional Cadaster Capabilities for Malaysia*. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XL-2/W4, 2015, pp. 53–55. <https://doi.org/10.5194/isprsarchives-XL-2-W4-53-2015>.
- [5] Karim H., Abdul Rahman A., Mohd Salleh M.R.: *Multi-scale and Scale Dimension Properties in Spatial Raster Modelling – Concept and Current Implementation*. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XLII-4/W16, 2019, pp. 339–345. <https://doi.org/10.5194/isprs-archives-XLII-4-W16-339-2019>.
- [6] Karim H., Rahman A.A., Boguslawski P.: *Generalisation Technique for 2D+Scale DHE Data Model*. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XLII-2/W1, 2016, pp. 61–67. <https://doi.org/10.5194/isprs-archives-XLII-2-W1-61-2016>.
- [7] Colucci E., De Ruvo V., Lingua A., Matrone F., Rizzo G.: *HBIM-GIS Integration: From IFC to CityGML Standard for Damaged Cultural Heritage in Multi-scale 3D GIS*. Applied Sciences, vol. 10(4), 2020, 1356. <https://doi.org/10.3390/app10041356>.

- [8] He S., Besuievksy G., Tourre V., Patow G., Moreau G.: *All range and heterogeneous multi-scale 3D city models*. Usage, Usability, and Utility of 3D City Models, 2012 02006. <https://doi.org/10.1051/3u3d/201202006>.
- [9] Breunig M., Borrmann A., Rank E., Hinz S., Kolbe T., Schilcher M., Mundani R.-P., Jubierre J.R., Flurl M., Thomsen A., Donaubaue A., Ji Y., Urban S., Laun S., Vilgertshofer S., Willenborg B., Menninghaus M., Steuer H., Wursthorn S., Leitloff J., Al-Doori M., Mazroobsemnani N.: *Collaborative Multi-Scale 3D City and Infrastructure Modeling and Simulation*. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XLII-4/W4, 2017, pp. 341–352. <https://doi.org/10.5194/isprs-archives-XLII-4-W4-341-2017>.
- [10] Pepe M., Costantino D., Alfio V.S., Angelini M.G., Restuccia Garofalo A.: *A CityGML Multiscale Approach for the Conservation and Management of Cultural Heritage: The Case Study of the Old Town of Taranto (Italy)*. ISPRS International Journal of Geo-Information, vol. 9(7), 2020, 449. <https://doi.org/10.3390/ijgi9070449>.
- [11] Gröger G., Kolbe T.H., Nagel C., Häfele K.: *OGC City Geography Markup Language (CityGML) Encoding Standard*. 2012.
- [12] Salleh S., Ujang U., Azri S.: *Virtual 3D Campus for Universiti Teknologi Malaysia (UTM)*. ISPRS International Journal of Geo-Information, vol. 10(6), 2021, 356. <https://doi.org/10.3390/ijgi10060356>.
- [13] Beil C., Ruhdorfer R., Coduro T., Kolbe T.H.: *Detailed Streetspace Modelling for Multiple Applications: Discussions on the Proposed CityGML 3.0 Transportation Model*. ISPRS International Journal of Geo-Information, vol. 9(10), 2020, 603. <https://doi.org/10.3390/ijgi9100603>.
- [14] Shen J., Zhou J., Zhou J., Herman L., Reznik T.: *Constructing the CityGML ADE for the Multi-Source Data Integration of Urban Flooding*. ISPRS International Journal of Geo-Information, vol. 9(6), 2020, 359. <https://doi.org/10.3390/ijgi9060359>.
- [15] Halim N., Chan K.: *SmartKADASTER Interactive Portal (SKiP), is it fit for purpose?* IOP Conference Series: Earth and Environmental Science, vol. 540, no. 1, 2020, 012025. <https://doi.org/10.1088/1755-1315/540/1/012025>.
- [16] Preka D., Doulamis A.: *3D Building Modeling in LoD2 using the CityGML Standard*. International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences, vol. 42, 2016. <https://doi.org/10.5194/isprs-archives-XLII-2-W2-11-2016>.
- [17] Biljecki F., Ledoux H., Stoter J., Vosselman G.: *The variants of an LOD of a 3D building model and their influence on spatial analyses*. ISPRS Journal of Photogrammetry and Remote Sensing, vol. 116, 2016, pp. 42–54. <https://doi.org/10.1016/j.isprsjprs.2016.03.003>.
- [18] Lei Tang L., Ying S., Li L., Biljecki F., Zhu H., Zhu Y., Yang F., Su F.: *An application-driven LOD modeling paradigm for 3D building models*. ISPRS Journal of Photogrammetry and Remote Sensing, vol. 161, 2020, pp. 194–207. <https://doi.org/10.1016/j.isprsjprs.2020.01.019>.

-
- [19] Ohori K.A., Biljecki F., Kumar K., Ledoux H., Stoter J.: *Modeling cities and landscapes in 3D with CityGML*. [in:] *Building Information Modeling*, Springer, 2018, pp. 199–215. https://doi.org/10.1007/978-3-319-92862-3_11.
- [20] Rahman A.A., Karim H., Jamali A., Buyuksalih G., Bayburt S.: *Conceptual Framework Towards Unified 3D Topological Modelling and Visualization Based on CityGML*. International Federation of Surveyors (FIG Congress 2018), Istanbul, Turkey, 2018.
- [21] Watt A., Eng N.: *Database Design – 2nd Edition*. BCCampus, Victoria, B.C. 2014. <https://opentextbc.ca/dbdesign01/>.
- [22] Kolbe T.H.: *Representing and exchanging 3D city models with CityGML*. [in:] Lee J., Zlatanova S. (eds.), *3D Geo-Information Sciences*, Springer, 2009, pp. 15–31.
- [23] Kutzner T., Kanishk Chaturvedi K., Kolbe T.H.: *CityGML 3.0: New Functions Open Up New Applications*. PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science, vol. 88, 2020, pp. 43–61. <https://doi.org/10.1007/s41064-020-00095-z>.