

3D CityGML BUILDING MODELS DEVELOPMENT WITH CROSS-SCALE QUERY DATABASE

Hairi Karim¹, Alias Abdul Rahman¹, Nur Zurairah Abdul Halim², Gurcan Buyuksalih³ and Hanis Rashidan¹

¹ 3D GIS Lab, Universiti Teknologi Malaysia, 81310 Johor Bahru, Malaysia.

² Department of Survey and Mapping Malaysia (JUPEM), 50578 Kuala Lumpur, Malaysia.

³ Istanbul, Turkey.

wmhairigis@gmail.com, alias@utm.my, nurzurairah@jupe.gov.my, gurcan.buyuksalih@gmail.com, mhanis27@gmail.com

Commission 4, WG 7

KEY WORDS: Multi-scale, CityGML LoD, City Modelling, 3DCityDB, PostgreSQL Database, Quality Control, Scale Unique ID, Cross-scale Query

ABSTRACT

CityGML model-based is now a 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 details (LoD, in version 2.0) of buildings. The LoDs are also known as pre-defined multi-scale models requiring a large storage-memory-graphic consumption than a single scale model. LoD CityGML models are primarily constructed using point cloud measurements and images of multiple systems, resulting in a range of accuracies and detailed model representations. Additionally, it entails several software, procedures, and formats for the construction of the respective LoDs prior to the final result in the CityGML schema. Thus, this paper discusses several issues of accuracy and consistency, proposing several quality controls (QC) for multiple data acquisition systems (e.g. airborne laser systems and mobile laser systems), model construction techniques (e.g. LoD1, LoD2, and LoD3), software (interchange formats), and migration to a PostgreSQL database. Additionally, the paper recommends the importance of minimising implementation errors. A scale-specific unique identifier is introduced to link all associated LoDs, enabling cross-LoD information queries within a database. Proper model construction, accuracy control, and format interchange of LoD models in accordance with national and international standards will undoubtedly encourage and expedite data sharing among data owners, agencies, stakeholders, and public users. A summary of the work and accomplishments is included, as well as a plan for future research on this subject.

1. INTRODUCTION

The paper describes the development of a 3D model from point clouds of several data acquisition techniques to CityGML version 2.0 schema database-ready for LoD0, LoD1, LoD2, LoD3 and LoD4. Overall processes from constructing 3D building models from point clouds to the database by using CityGML schema. The point clouds were captured from Airborne Laser System (ALS) and Mobile Laser System (MLS). It involves several techniques for respective LoDs, software, formats, quality checking and database. This paper aims to share our research and project experiences in designing and handling the models for the cadastre domain. CityGML is an international standard by the Open Geospatial Consortium (OGC) for spatial representation and exchange of 3D city models. It defines the three-dimensional geometry, topology, semantics, and appearance of the most relevant topographic objects (e.g. building structures) in urban areas, as reported by Jovanović et al. (2020). Several research studies on CityGML multi-scale have been carried out by several researchers (Colucci et al., 2020; Breunig et al., 2017; He et al., 2012), including urban changes of Taranto (simulation from 1800) by Pepe et al. (2020). Some works on 3D CityGML buildings modelling (LoD1 and LoD2) was automatically constructed from LiDAR point clouds data by Jayaraj and Ramiya (2018) and Büyüksalih et al. (2019). However, there were no detailed discussions on the quality control of the models. Besides, available publications on 3D building construction did not describe in detail some technical workflows, matters to be concerned as for potential errors and limitations of the embarked solutions. On the other hand, only a few publications

and guidelines on CityGML implementation using database implementation (e.g. PostgreSQL), especially from 3DCityDB and Yao et al. (2018). Reports on real implementation experiences are hardly available. Besides, best practices and potential errors during model construction, format interchange, database migration and assessment of the database are hardly discussed in other research publications – thus, no guidelines for new real implementation for multi-scale 3D building city models.

Throughout this paper, quality control (QC) will be found in several sections – with a different meaning. QC in 3D building construction modelling refers to several accuracy controls at merging ALS and MLS point clouds, completeness of façade texture images and sketching the model based on measured point clouds (e.g. $\pm 0.3\text{m}$ for this project/domain, but generally in $\pm 2\text{m}$ for LoD2 and $\pm 0.5\text{m}$ for LoD3 according to CityGML standards). However, the quality assurance (QA) term will mainly be used for process workflow and QC to prevent mistakes on each process migrations and final model. Later in Section 4 and 5, QC refers to the model interoperability format exchanges, migration process output (no missing, misplace or duplicate), standards compliance (scale unique ID and CityGML classes), subclass and texture quality.

The remaining of the paper is as follows: Section 2 describes CityGML model construction and Section 3 deals with potential errors for QC during the modelling phase. While Section 4 introduce scale unique ID, QC, and database migration. Section 5 describes on database assessment on supporting cross-scale and finally concluding remarks in Section 6.

2. 3D BUILDING MODELS CONSTRUCTION BASED ON POINT CLOUDS

This section is divided into three sub-sections for point clouds building preparation (ALS and MLS), construction of LoD0-LoD1 and lastly, construction of LoD2-LoD3.

2.1 Point Clouds for Building Model

Dealing with a substantial point clouds data costs significantly in high-end workstation specifications (high performance). Individual ALS and MLS data typically consumed a lot of graphic and workstation memory, especially to load, process and construct 3D models from the point measurement. One of the best ways to expedite the construction of 3D model is to subdivide the point cloud to the respective building block with only a single high-end workstation. The process is known as “clipping” to get a smaller area and then save it as a new dataset (Table 1) using LiDAR360 software. It will greatly speed up the 3D construction process (parallel) with multiple lower specification computers (cost-effective) for the LoD2, LoD3 and LoD4 building modelling purposes. Later, the clipped ALS and MLS selected building files should be merged into a single file for the purpose of a 3D model sketching (Figure 1) in SketchUp or other 3D modelling software.

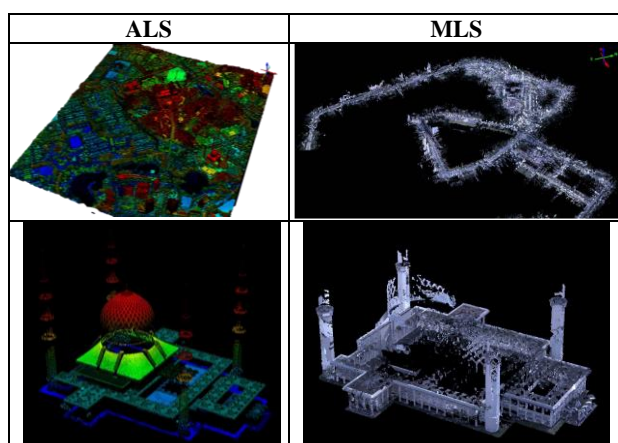


Table 1. Example of coloured LiDAR point clouds from ALS and MLS datasets and selection of single building for LoD2 and LoD3 modelling.

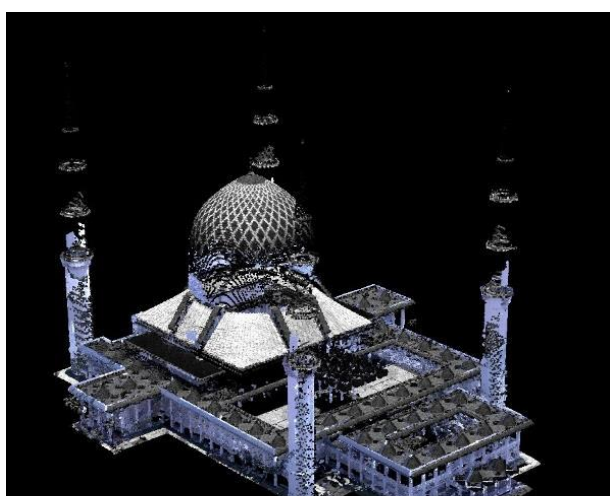


Figure 1. Combination of selected ALS and MLS LiDAR data for a building (e.g., a mosque).

2.2 LoD0 and LoD1

LoD0 is a footprint of digitized manually from orthophoto images (captured during ALS mission) as in Figure 2. Automatic extraction is not the best option since the study area mainly comprises tree canopies overlapping the structure of building footprints. However, some of the project areas without tree canopies and with clear building structures (boundary) are extracted automatically using the available ArcGIS function described by Chafiq et. al, (2021). Later, a new column is added into the building footprint attribute layer for assigning ID supporting LoD. Malaysian’s cadastre Unique Parcel Identification ID (UPI ID) was extended to support 3D Scale Unique ID with multiple LoDs from LoD0-LoD4 as D0-D4 (Halim et. al., 2021). The scale unique ID was extracted automatically from cadastre lot 3D UPI ID with an extension of scale ID as a new attribute column for each LoD0 building footprint polygon. Example of scale unique ID as discussed in Section 4.1.

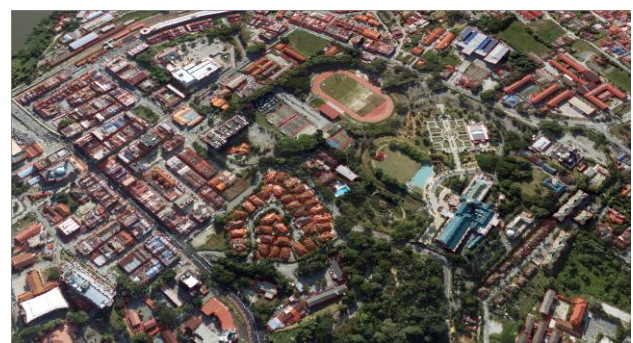


Figure 2. Building footprints (LoD0) in red polygon as digitizing result from orthophoto.

The further process is to construct LoD1. FME workbench was chosen as a practical implementation for automatic extrusion technique for constructing LoD1 from LoD0. The input layer will be the LoD0 (footprint with ID) and the filtered ALS point clouds (only buildings), as shown in the script below (Figure 3). The study uses the mean value of the rooftop point cloud to generate a flat LoD1 model of each rooftop surface. Thus, each building has its own height level. The scale unique ID of LoD0 will be transferred into LoD1 ID with some additional code changing the extension of D0 to D1 for LoD1 automatically as in the FME workflow below (Figure 3).

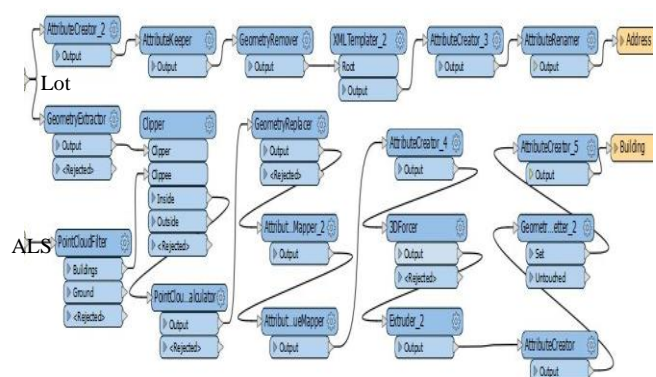


Figure 3. FME workbench model builder for extrusion of CityGML LoD0 (building footprint) to LoD1 using point cloud data and scale unique ID.

After the FME process, the results of LoD1 buildings (Figure 4) with scale unique ID embedded into attribute table for each model. These LoD1 models are basically ready for migration after one-to-one building block quality check using FME Data Inspector, especially on the three aspects:

- LoD1 footprint and LoD0 is 100% matched.
- Rooftop is in between building rooftop's point clouds.
- Correct scale unique ID for LoD1 (D1).



Figure 4. Transformation models from LoD0 (footprint, first image from top) and LoD1 (second image).

2.3 LoD2 and LoD3 Models Construction

On the other hand, LoD2, 3 and 4, do not have any automatic conversion or transformation process, and thus need manual measurement and construction process within a few available 3D software such as Revit and Google SketchUp. As for our work, we chose to construct the 3D model with SketchUp since it supports actual coordinate (spatial local coordinate) and less complicated GUI as compared to Revit software (mainly for 3D building of Building Information Modelling, BIM).

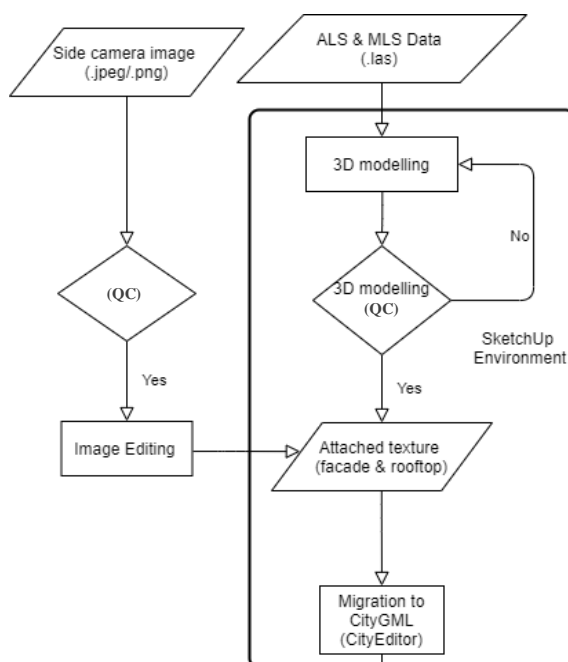


Figure 5. LoD 2, 3 and 4 model construction workflow using point cloud data and SketchUp software.

Figure 7 summarises overall construction process of LoD0-LoD3 CityGML building models of this section. Our work

utilises LiDAR360 software for clipping purposes and two software (FME and SketchUp) for construction of 3D building models according to CityGML standards as for minimizing the software cost.

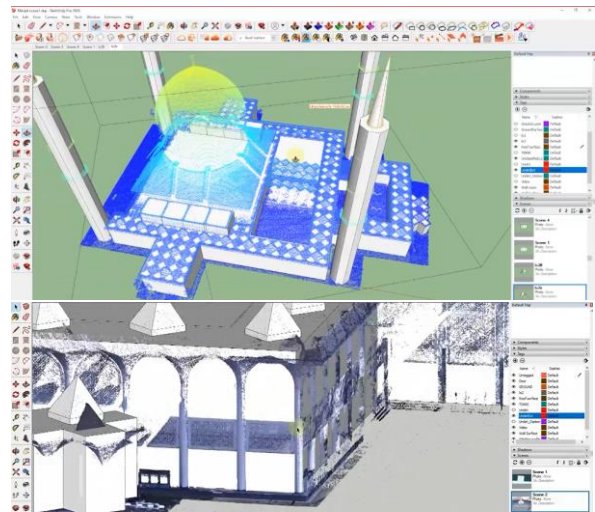


Figure 6. Constructing LoD2 and LoD3 from rooftop to building façade using SketchUp software.

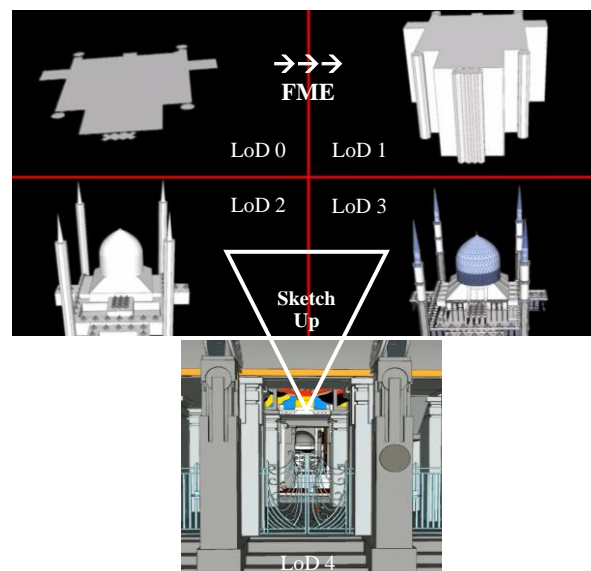


Figure 7. Summary of CityGML LoDs construction based on this works.

3. ERRORS DURING CONSTRUCTION OF BUILDING MODELS

3.1 ALS Point Clouds

The main concern on the ALS is the point clouds density (number of points per meter square). It is imperative to have a good number of point densities, especially for LoD3 and LoD4, as the accuracy level increases (detailed model). Good numbers refer to the structure of the building. For example, a flat rooftop building has a minimum of 8 points/m², and a complex rooftop requires a higher point density.

The second concern is establishing localized Ground Control Points (GCPs) for cross-checking with collected point clouds

(ALS and MLS) for each building to be modelled. The GCP used in this project is based on the static positioning technique (coordinate) and is tied with three good cadastre boundary stones. However, it may be different from other domain requirements. It is also to ensure that the height of the point cloud at the rooftop with the ground (e.g. building façade) by other datasets such as the MLS dataset are interconnected. Common control points should be established from the nearest GCP to prevent gaps (e.g., Figure 8) and overlaps between these datasets for LoD3 and LoD4 model constructions.

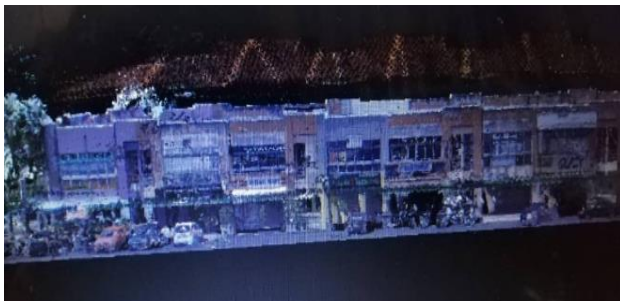


Figure 8. Example of a “gap” error in between ALS and MLS point cloud datasets to be controlled.

3.2 MLS point clouds data

Raw clipped MLS and ALS point clouds data for each building should be checked before merging into a new file (Figure 1, previous Section 2.1). For example, ALS data and MLS mission track of the same grid should be compared with GCP located within the same grid (localized GCP). Later, QC should double-check before clipping point clouds to the selected building, merging ALS-MLS into a single point clouds file (for 3D model construction). This QC should be done in a high specification machine (workstation) to minimise potential errors (due to data collection and pre-processing as in Figures 9-11 for the 3D building construction phase (sketching model).



Figure 9. Example of MLS error - duplicate point clouds data of the same building with an axis shifted. They have potentially accrued in pre-processing of raw data duplicated moved as manual correction toward GCP.



Figure 10. Example of MLS error - incompleted point clouds data potentially due to the vehicle route very close to the ground building canopy.

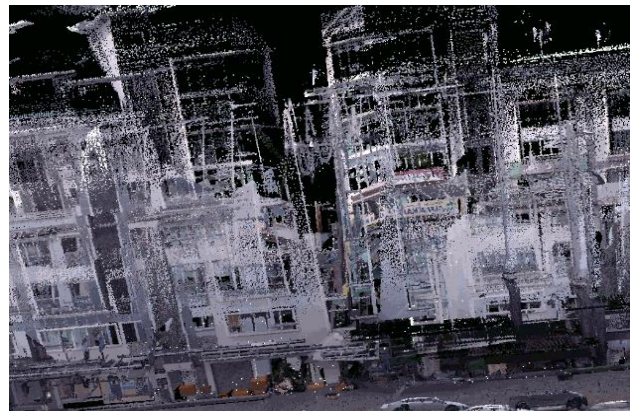


Figure 11. Example of MLS error - tilted building point clouds due to the road bumpers and holes at data collection phase.

Apart from the point clouds, mobile mapping also comes typically with raw photos of six side cameras/angles for 360 mapping purposes (e.g. Leica Pegasus system). Selected buildings to be modelled requires façade textures extracted from 360 side camera's raw images in each mission track. However, some potential limitations may arise as the building façade images are blocked by road furniture, other vehicles, and trees (e.g. Figure 12). Thus, selection of route and data collection time should be considered during MLS data acquisition survey. These problems will increase editing time, and workload for the modelling as the façade texture needs to be edited manually (e.g. in Photoshop software) to enhance image quality.



Figure 12. Example of blockers/noises for the building façade texture from raw 360 side cameras.

4. PREPARATION OF MODELS TOWARD DATABASE READY (MIGRATION).

For each LoD1-LoD3, a building block needs to be split into their respective units accordingly based on 2D cadastre lot boundary or coloured point clouds (shared wall). This ensures that more details information could be stored individually to a

higher LoD level (e.g. business signboard and company name for each unit at LoD3). Thus, a unique scale ID will be introduced and key-in for 3D unit models (individual splitted model) of a building block for respective LoD1, LoD2, LoD3 and LoD4 (if any) to their respective object classes (Table 2 in Section 4.3). The unique scale ID and individual building block unit are two primary components supporting cross-scale query (cross-LoD information retrieval) later in this paper. Splitting a building block to each respective unit space boundary (Figure 13) should be applied for each LoDs. There are two efficient methods to perform it based on respective LoDs and software that we used:

- Split LoD0 (building footprint) as per respective cadastre lot and extrude them using FME workbench for LoD1
- S4u Slice extension tool (yearly licensed subscription) in SketchUp for LoD2, LoD3 and LoD4.

The workflow and research methodology in this study/project implementation is illustrated in Figure 14. It is based on the proposed concept of a single layer in a single viewer from our previous publications, Karim et. al (2018) and Rahman et. al. (2018). The previous sections mainly describes the second phase of this research methodology on perfect CityGML model construction and scale unique ID before moving toward the third phase of a database environment. However, in this paper,

we highlight the potential errors that will arise for each phase and thus the proposed QCs will be introduced for each phase and transition between two consecutive phases (QC as green start symbol in Figure 14) for quality assurance. While, the third phase of this methodology is described in Section 5 as a database environment (migration using 3DCityDB tool, QC and cross-scale information query).

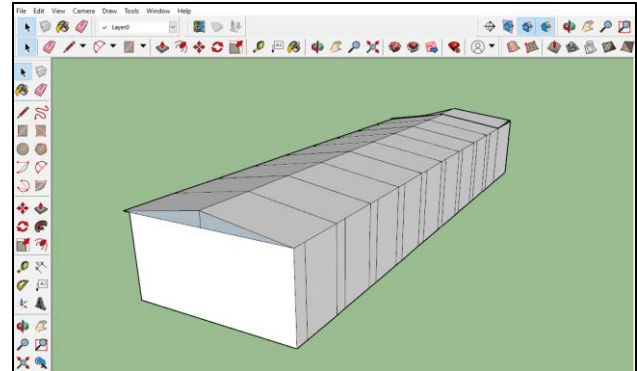


Figure 13. Result of splitting a building into respective units (e.g. commercial lots) for LoD2 using SketchUp software.

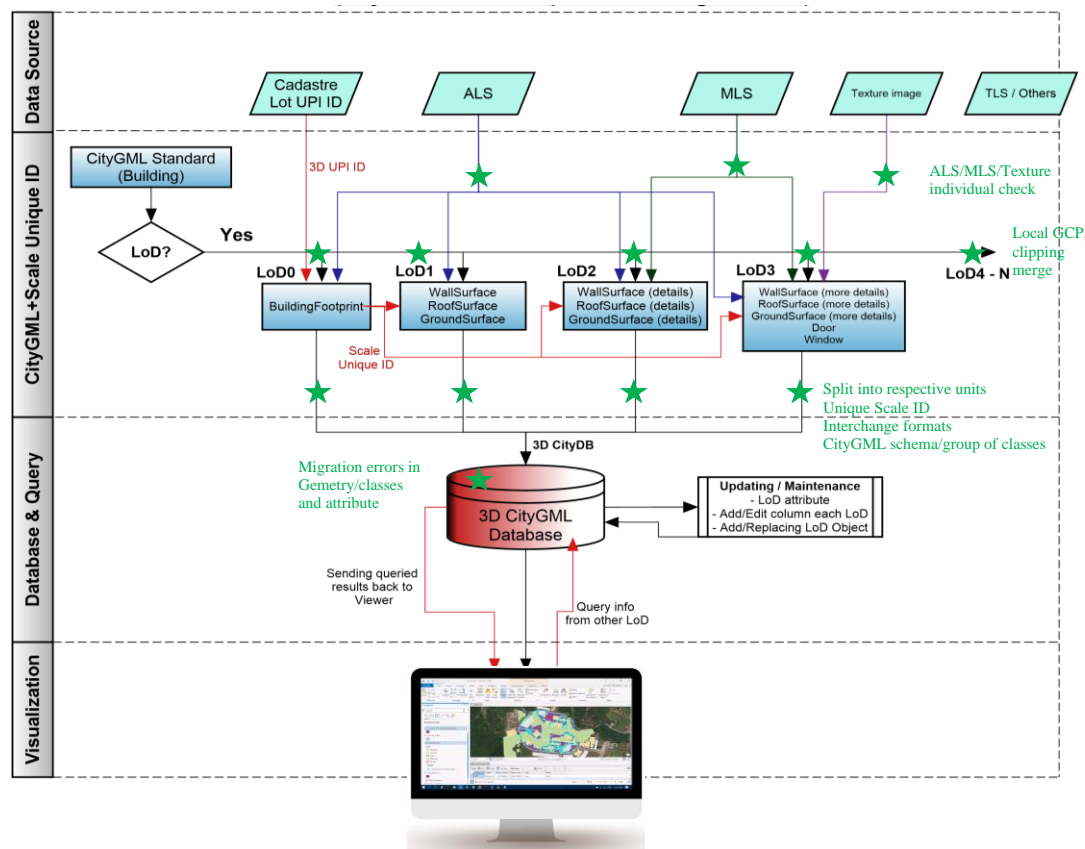


Figure 14. Overall general research methodology/project implementation workflow with stages of QC check.

4.1 Assigning Scale Unique ID for Each Multi-scale Model

Assigning a unique ID for each 3D building is compulsory since we need to extend the object/building model ID to support cross-scale LoDs queries. This is to ensure easiness to retrieve the specific building information in that particular cadastre lot and the sub-classes of building groups (e.g. wall, window, door, building installation, and others). For instance, a scale unique

ID is UPI_10010100031488.S.0B.M1.D3, where (S) is referred to strata unit, M is the number of buildings per lot, and D3 is the CityGML LoD3. Further details on this process can refer to section 4 of the published document by Halim et. al. (2021). The paper described the assigning of Unique Scale ID for the database in the 3D Cadastre perspective domain. The assignment of the ID and building group classes is done in SketchUp software at modelling phase (Figure 15). Later, after

converting SKP to GML format using FME workbench, FME Data Inspector is used in QC for any missing ID, geometry and CityGML sub-object (group) classes as in Figure 16.

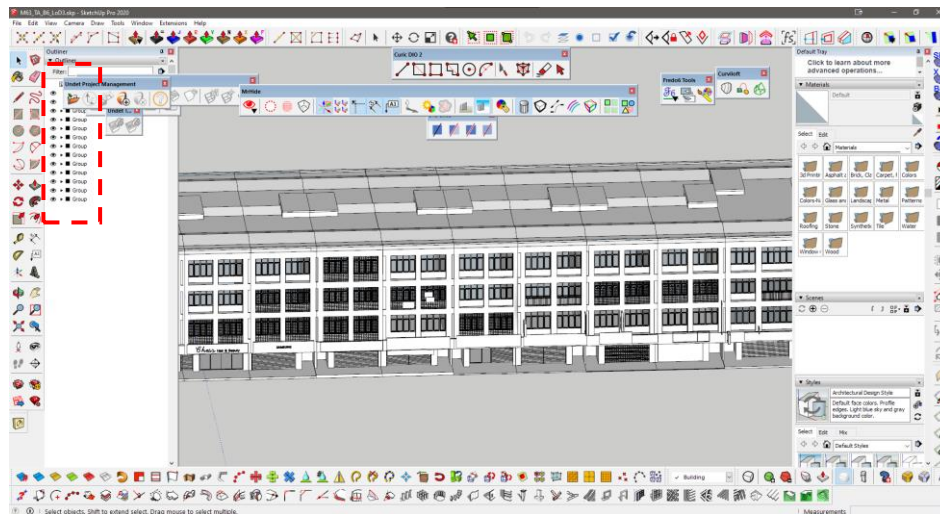


Figure 15. Importing SketchUp to CityGML model (LoD1) while defining 3D ID and group classes (red dotted box) for the model.

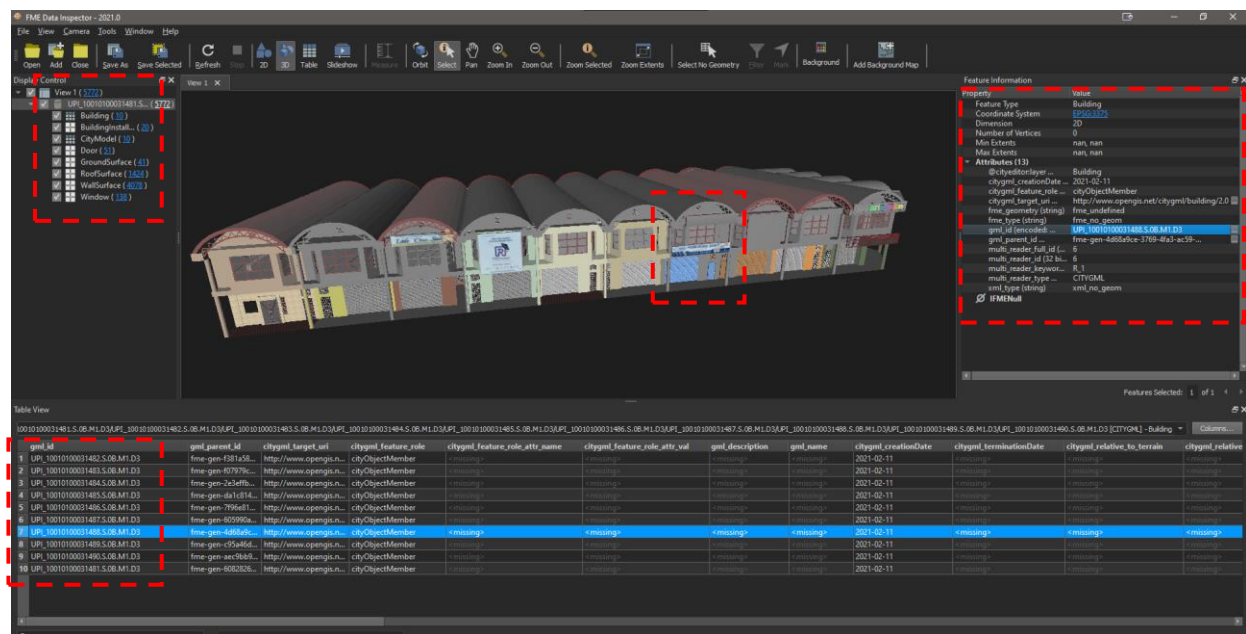


Figure 16. QC checking process for any missing sun-classes objects (signboard, texture and complex geometry) and unique scale ID in XML format using FME Data Inspector .

Before migrating the model to PostgreSQL database using 3DCityDB, quality control and checking are conducted on the scale unique ID and the model's geometry suited to the CityGML schema.

4.2 Quality Check and Format Interchange using FME

Overall format interchange and QC model from SkechUp format to PostgreSQL are illustrated in Figure 17. FME Data Inspector is used to check the model completeness and complying CityGML standards (e.g. Figure 18 – checking any overlap/gaps during XML conversion). While FME Quick Translator for format interchange is utilized to convert XML format (save as in SketchUp software) to GML format as the requirement of 3DCityDB format.

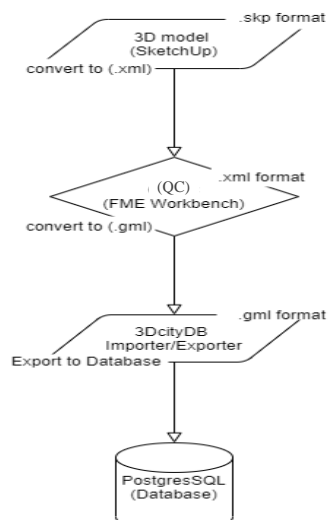


Figure 17. 3D model migration workflow from model construction in SketchUp to PostgreSQL database.

4.3 Classification of 3D Model Sub-classes According to CityGML Schema (database ready)

Dealing with multiple LoD, especially in CityGML, each LoD requires specific additional details and classes from the standard 3D model. Utilizing 3DCityDB importer/exporter helps to double-check, improve the quality control and standard for CityGML group schema as listed in Table 2. Previously, these classes are declared at SketchUp software.

Layers Group\LoD		LoD1	LoD2	LoD3	LoD4
1	Solid				
2	Building (Group)				
3	Wall Surface				
4	Roof Surface				
5	Ground Surface				
6	Door (Group)				
7	Window (Group)				
8	Building Installation (Group)				
9	Interior Building Installation (Group)				
10	Building Furniture (Group)				
11	Closure Surface				
12	Floor Surface				
13	Ceiling Surface				
14	Room (Group)				

Table 2. Group layer classification (classes) for each CityGML LoD in 3DCityDB.

After the quality check and any necessary correction on the scale unique ID, group classes for each CityGML LoD, now the models are ready to be imported to PostgreSQL database.

5. DATABASE MIGRATION, QC AND CROSS-SCALE QUERY

As previously mentioned, migration of CityGML file (GML format) utilises 3DCityDB tool into PostgreSQL database. Once the first migration is completed, 3DCityDB creates CityObject

schema tables for CityGML models. One of the generated tables is Building schema that stores 3D building models and information in all LoDs.

Depending on the number of buildings and their complexity, the tool can migrate the models in bulk or by parts. For example, LoD1 can be imported by a grid of 6.25km², while LoD2 and LoD3 are migrated by MLS mission (e.g. 10 selected buildings in 5km mission length). However, LoD4, which is very heavy (rendering), is imported by group classes of a building model (worth 25Gb file storage). Figure 18 shows an example of a migration process log, and Figure 19-22 show the cross-check of numbers of each migrated LoDs. Numbers of imported group classes and object for each LoDs should be counted before and after the migration. For instance, 40 models are imported (input) in 3DCityDB and the same number of models need to confirm success in migration log report (Figure 18) and query (by time, as in Figure 21) within database table.

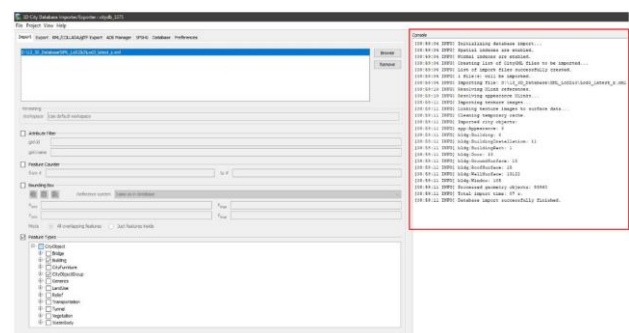


Figure 18. An example of several LoD3 buildings with texture model successfully imported into 3D database (red box is the imported log report).

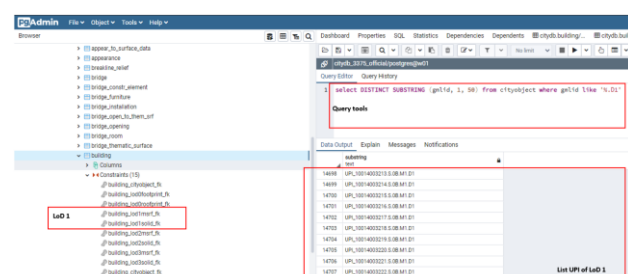


Figure 19. Scale Unique ID for LoD1.

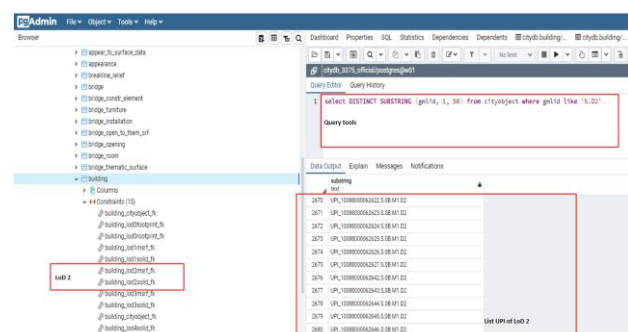


Figure 20. Scale Unique ID for LoD2.

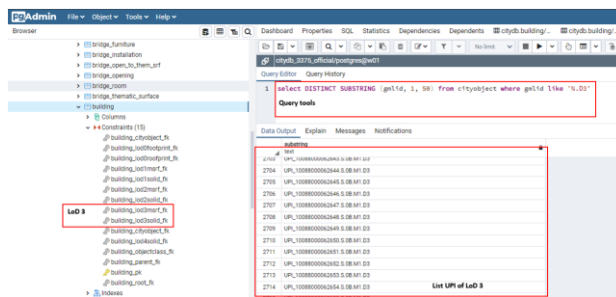


Figure 21. Scale Unique ID for LoD3.

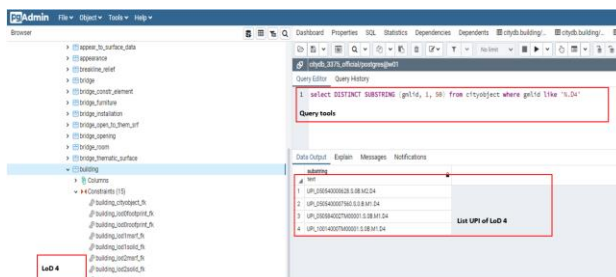


Figure 22. Scale Unique ID for LoD4.

5.1 QC of Migration and Final Model

Migrated models inside the database should be one-to-one cross check for any errors during the migration process. Several aspects for consideration and QC include but not limited to:

- Coordinate system for each LoD.
- Each LoD model must have the same based heighting coordinate (e.g. LoD1, LoD2, LoD3 and LoD4 if any)
- Texture and texture ID also migrated along the model
- Scale Unique ID for each LoD
- Correct geometry and it group classes (Table 2).



Figure 23. The migration is not 100% guaranteed to be correct. Double-check should be done for error detection as in the window class in this figure.

Apart from the model accuracy and CityGML standards for each QC level discussed in previous phases and sections, the generated model also can be evaluated as in the final result, including texture. For example, Figure 24 shows a comparison of reality on-site with the sketched model in LoD3.



Figure 24. An example of comparison between on-site (reality) and the generated model.



Figure 25. An example of correct geometry (splitted by lot), scale unique ID and .texture for each building unit.

5.2 Other Aspects for Quality Consideration

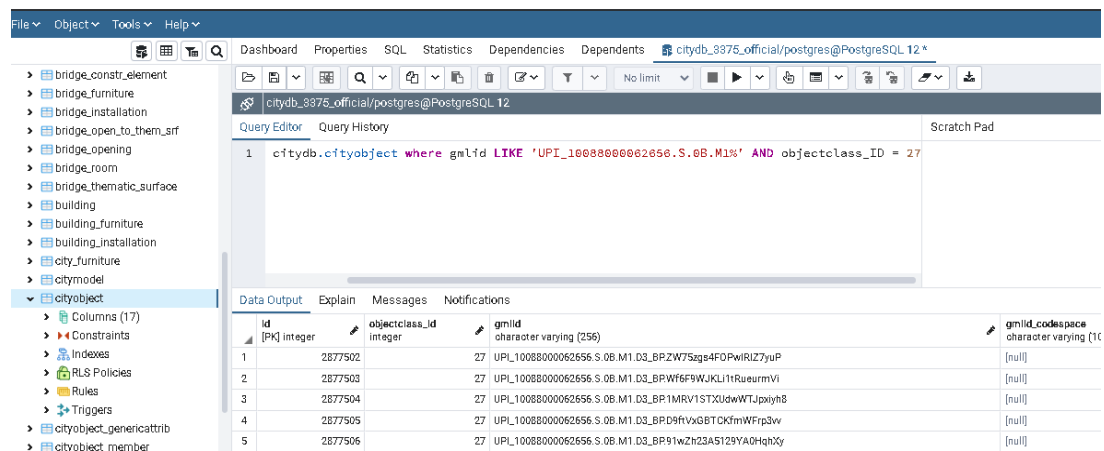
The following are some recommendations for ensuring a quality 3D construction:

- Combining LiDAR from multiple sources such as ALS, MLS, and Terrestrial Laser System (TLS) for LoD4 should consider local GCP (closed to the building). It ensures the ground (Digital Elevation Model, DEM) is connected with all point clouds from multiple systems.
- Each selected software to be used in sketching 3D building, for interchange format and quality check, provides different decimal points and coordinate system parameters. Thus, for consistency of the measurement and quality, please ensure all correct settings on each software meeting project requirements.
- Conducting QC with different operators to minimize human error (e.g. 3D sketching by personnel A, and B for QC of the same model).

5.3 Cross-scale Information Query

The term cross-scale query refers to capability to conduct information retrieval from other LoDs layer (attribute) by using query syntax in database or developed system especially map viewer. The query is possible to perform since we already introduced scale unique ID in each LoDs model. Later, this capability able to view low detail level (e.g. LoD1 or LoD2) in viewer, however details information on the building such as number of window, door, floor, table, chairs, name, installation (e.g. air-conditioner etc) could be called out from the database. Example of cross-scale information retrieval are as follow (e.g. Figure 26 and Figure 27):

```
select * from citydb.cityobject where gmlid LIKE 'UPI_10088000062656.S.0B.M1%' AND objectclass_ID = 27
```

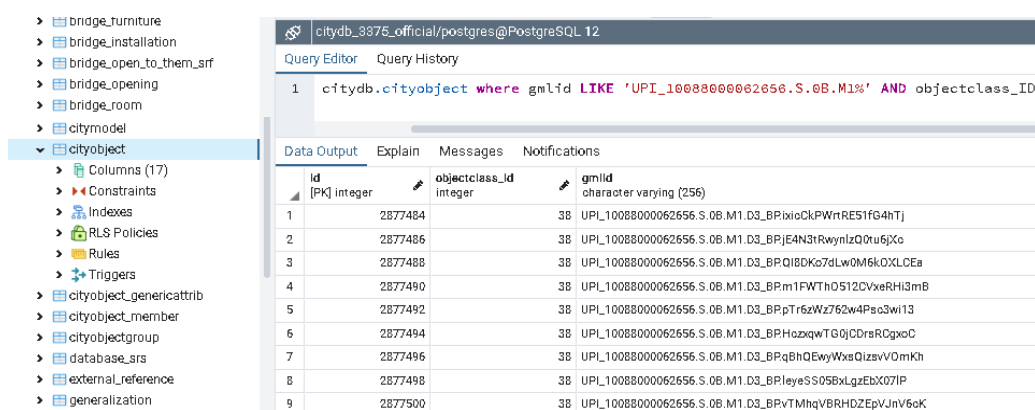



Query Editor Query History Scratch Pad

```
1 citydb.cityobject where gmlid LIKE 'UPI_1008800062656.S.0B.M1%' AND objectclass_ID = 27
```

	id [PK] integer	objectclass_id integer	gmlid character varying (256)	gmlid_codespace character varying (16)
1	2877502	27	UPI_1008800062656.S.0B.M1.D3.BPZW7Sg4F0PwIRIZ7yUP	[null]
2	2877503	27	UPI_1008800062656.S.0B.M1.D3.BPWf6F9WJL1tRueurmVi	[null]
3	2877504	27	UPI_1008800062656.S.0B.M1.D3.BP1MRV1STXUdwWTJpxiuh8	[null]
4	2877505	27	UPI_1008800062656.S.0B.M1.D3.BPD9frv3G8BTCKfmWFr3v	[null]
5	2877506	27	UPI_1008800062656.S.0B.M1.D3.BP91wZ23A5129YA0HqHxY	[null]

Figure 26. Example building UPI_1008800062656.S.0B.M1 having 5 doors (object class 27) in LoD3 representation; zero in LoD2.



Query Editor Query History

```
1 citydb.cityobject where gmlid LIKE 'UPI_1008800062656.S.0B.M1%' AND objectclass_ID
```

	id [PK] integer	objectclass_id integer	gmlid character varying (256)	gmlid_codespace character varying (16)
1	2877484	38	UPI_1008800062656.S.0B.M1.D3.BPjioCkPWrtRES1FG4hTj	[null]
2	2877486	38	UPI_1008800062656.S.0B.M1.D3.BPJ4N3trWynlZQ0tu6jXc	[null]
3	2877488	38	UPI_1008800062656.S.0B.M1.D3.BPJ8DKo7dLw0M6kVXLCeA	[null]
4	2877490	38	UPI_1008800062656.S.0B.M1.D3.BPm1FWTth0512CVxeRHl3mB	[null]
5	2877492	38	UPI_1008800062656.S.0B.M1.D3.BPTr6zWz762w4Psc3wi13	[null]
6	2877494	38	UPI_1008800062656.S.0B.M1.D3.BPHozxqWGTG0jCDrsRCgxcC	[null]
7	2877496	38	UPI_1008800062656.S.0B.M1.D3.BPqBhQEwyWxsQizsvV0mKh	[null]
8	2877498	38	UPI_1008800062656.S.0B.M1.D3.BPleyeS05BxLgZEBX07IP	[null]
9	2877500	38	UPI_1008800062656.S.0B.M1.D3.BPvTMhqvVRHDZEpJvnV6oK	[null]

Figure 27. Example building UPI_1008800062656.S.0B.M1 having 9 windows (object class 38) in LoD3 representation.

6. CONCLUSION

This paper describes the construction of LoDs 3D buildings in the CityGML Version 2.0 schema, as well as the assessment of the quality and consistency of format interchange/migration to the PostgreSQL database. The quality controls are measured in several levels especially during data source checking, combining several datasets, model construction (according to CityGML LoDs and client requirements), format interchanges between software and migration into database. Also, in this paper, the scale unique ID is introduced as one of the techniques for enabling cross-scale information queries for single viewer readiness. Updating attributes for each LoD becomes much easier and less expensive (with less storage and less maintenance). This study demonstrates that each correctly defined LoD model can be linked, managed more effectively, and work properly toward a single model in a single viewer (using a cross-scale query). As a result, the cost of multi-scale model maintenance, time, and viewing machine specifications (computer) could be significantly reduced.

This work's results could be used as guidelines for others (vendor of 3D modeller, software providers, users or system developers), particularly for smart city modelling and data sharing. We intend to focus our efforts in the future primarily on a single visualisation platform, as the existing platform mostly does not support cross-scale information queries.

ACKNOWLEDGEMENTS

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

REFERENCES

- Breunig, M., Borrmann, A., Rank, E., Hinz, S., Kolbe, T., Schilcher, M., Mundani, R. P., Jubierre, J., Flurl, M., Thomsen, A., Donaubauer, A., Ji, Y., Urban, S., Laun, S., Vilgertshofer, S., Willenborg, B., Menninghaus, M., Steuer, H., Wursthorn, S. & Mazroob Semnani, N. 2017. Collaborative Multi-Scale 3D City and Infrastructure Modeling and Simulation. ISPRS -XLII-4/W4, 341-352.
- Büyüksalih, G., Başkaraca, P., Rahman, A. A., 2019. 3D City Modelling of Istanbul. Issues, Challenges and Limitations. In GIS International Article, <https://www.gim-international.com/content/article/3d-city-modelling-of-istanbul>
- Chafiq T., Hachimi H., Raji M., Zerraf S. (2021) U-Net: Deep Learning for Extracting Building Boundary Collected by Drone of Agadir's Harbor. In: Motahhir S., Bossoufi B. (eds) Digital Technologies and Applications. ICDTA 2021. Lecture Notes in Networks and Systems, vol 211. Springer, Cham. https://doi.org/10.1007/978-3-030-73882-2_11

Colucci, E., Ruvo, V. D., Lingua, A., Matrone, F. & Rizzo, G. 2020. HBIM-GIS Integration: From IFC to CityGML Standard for Damaged Cultural Heritage in Multiscale 3D GIS. *Applied Sciences* 10.

Halim, N. Z, Lim, C.K., and Karim, H., 2021. Developing a 3D City Model Database Beyond Cadastral Purposes. GGT 2021, Malaysia. IOP Conf. Ser.: Earth Environ. Sci. 767 012027

He, S., Besuievksy, G., Tourre, V., Patow, G. & Moreau, G. 2012. All range and heterogeneous multi-scale 3D city models.

Jayaraj, P., and Ramiya, A. M., 2018. 3D Citygml Building Modelling from LiDAR Point Cloud Data. ISPRS Volume XLII-5, 2018. ISPRS TC V Mid-term Symposium “Geospatial Technology – Pixel to People”, 20–23 November 2018, Dehradun, India.

Karim, H., Rahman, A., & Jamali, A. (2018). Unified Topological Framework for Retrieving 2d and 3d Multi-Scale Spatial Information. ISPRS - 4249, 265-272.

Pepe, M., Costantino, D., Alfio, V. S., Angelini, M. G. & Restuccia Garofalo, A. 2020. 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.

Rahman, A., Karim, H., Malaysia, Gurcan, & Bayburt, S. (2018). Conceptual Framework Towards Unified 3 D Topological Modelling and Visualization Based on CityGML.

Smith, J., 2000. Remote sensing to predict volcano outbursts. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XXVII-B1, 456-469.

Yao, Z., Nagel, C., Kunde, F., Hudra, G., Willkomm, P., Donaubaue, A., Adolphi, T., and Kolbe, T.H., 2018. 3DCityDB - a 3D geodatabase solution for the management, analysis, and visualization of semantic 3D city models based on CityGML. Springer, DOI: 10.1186/s40965-018-0046-7.