

2013-6

## Preparing Detailed 3D Building Models for Google Earth Integration

Linh Truong-Hong

*Technological University Dublin, linh.truonghong@tudublin.ie*

Thanh Thoa Pham Thi

*Technological University Dublin, thoa-pham@tudublin.ie*

Junjun Yin

*Technological University Dublin, junjun.yin@tudublin.ie*

*See next page for additional authors*

Follow this and additional works at: <https://arrow.tudublin.ie/dmcccon>



Part of the [Databases and Information Systems Commons](#)

---

### Recommended Citation

Linh Truong-Hong et al. (2013) Preparing Detailed 3D Building Models for Google Earth Integration: B. Murgante et al. (Eds.): *ICCSA 2013*, Part IV, LNCS 7974, pp. 61–76. Springer, Heidelberg.

This Conference Paper is brought to you for free and open access by the Digital Media Centre at ARROW@TU Dublin. It has been accepted for inclusion in Conference papers by an authorized administrator of ARROW@TU Dublin. For more information, please contact [arrow.admin@tudublin.ie](mailto:arrow.admin@tudublin.ie), [aisling.coyne@tudublin.ie](mailto:aisling.coyne@tudublin.ie), [vera.kilshaw@tudublin.ie](mailto:vera.kilshaw@tudublin.ie).

Funder: Science Foundation Ireland

---

**Authors**

Linh Truong-Hong, Thanh Thoa Pham Thi, Junjun Yin, and James Carswell

# Preparing Detailed 3D Building Models for Google Earth Integration

Linh Truong-Hong, Thanh Thoa Pham Thi, Junjun Yin, and James D. Carswell

Digital Media Centre, Dublin Institute of Technology, Ireland  
{linh.truonghong, thoa.pham, jcarswell}@dit.ie,  
yinjunjun@gmail.com

**Abstract.** Today's spatially aware users are becoming more interested in retrieving personalised and task relevant information, requiring detailed 3D city models linked to non-spatial attribute data. However, current implementations of 3D city models are typically LoD2 that don't include geometric or attribute details about many visible features (e.g. rooms) of a building. As such, value-added applications developed for web-based and wireless platforms are limited to querying for available non-spatial business data at the building level only. To overcome this, geometrically accurate 3D building models are necessary to enable users to visualize, interact, and query for task specific non-spatial business data. This paper proposes a workflow for creating detailed 3D building models with LoD3 from TLS data and uploading these models into Google Earth so that users can then explore the non-spatial business data of a building and its sub-components (e.g. windows, doors, rooms). Processing bottlenecks of the proposed workflow for detailed 3D building reconstruction are also discussed.

**Keywords:** LiDAR point clouds, 3D building models, spatial business data, KML, Google Earth.

## 1 Introduction

The need for three-dimensional (3D) geo-information of a city is growing and expanding rapidly in a variety of research fields since spatial data is essential for a large range of applications (e.g. environmental planning and monitoring, disaster management, security, telecommunications, location-based services). These applications provide public access to explore 3D geographic objects such as buildings, roads and other built environment infrastructure, of which the 3D building models are our main interest.

The raw data used to reconstruct detailed 3D building models for 3D city maps is obtained from various resources through a wide range of techniques. With recent developments in photogrammetry and remote sensing, building models can be automatically reconstructed given the geometric resolution of satellite imagery and Light Detection and Ranging (LiDAR) point cloud data [1]. Geometric and semantic properties of 3D models are typically stored in five consecutive levels-of-detail

(LoD), in which LoD0 defines a coarse regional scale model while LoD4 denotes architectural building models with detailed walls, roof structures, balconies, interior structures, and detailed vegetation and transportation objects. An advantage of the LoD approach is the coherent modeling of semantics and geometric/topological properties together at each level, where geometric objects get assigned to semantic objects.

Contemporary approaches appear limited to reconstructing 3D building models at level-of-detail (LoD) 2, which incorporates external surfaces such as walls and roofs of a building only [1, 2]. As today's spatially aware users are becoming more interested in task relevant information, for example, users may want to query a room's name or a room's purpose in a public building (e.g. government office, school, or hospital), they require more geometrically accurate and spatially detailed building models in order to query and retrieve such location specific attributes. Therefore, building models with higher level-of-detail (either LoD3 or LoD4) are required for such task specific interaction with 3D city models. However, automatically producing such detailed 3D models is still a big challenge because of the complexity of the scenes and the time consuming data collection and manual model processing involved.

Detailed 3D city models can provide a more realistic geographical representation of an entire city enabling users to retrieve specific object information when interacting with the model. At present, there are various mapping tools available for creating a 3D city based on computer-aided design (CAD), Virtual Reality Markup Language (VRML), or Keyhole Markup Language (KML) architectures. While these systems can handle 3D spatial data and associated (non-spatial) attribute or business data, current implementations of 3D cities only allow users to obtain general information about its objects, for example, the name or address of a building. Exploring non-spatial business data of sub-objects like windows and doors (and their associated rooms) to retrieve the purpose, content, or schedule of a specific room in a building is still an evident limitation of existing web-based Geographic Information System (GIS) applications. This is largely due to insufficient geometric detail of online building models integrated with today's web mapping platforms, and subsequently a general lack of spatially linked business data.

This paper proposes a workflow for generating detailed 3D building models at LoD3 from LiDAR data and subsequently visualizing these building models in a web-mapping platform where users can explore non-spatial attribute data of not only the building but also its sub-components (e.g. rooms). Additionally, processing bottlenecks of current techniques for converting raw LiDAR data to detailed 3D building models for real-time display in Google Earth (GE) is also discussed.

The remainder of this paper is organized as follows. The related work section focuses on building reconstruction from LiDAR data and virtual 3D city models. Following this, a workflow is proposed for reconstructing building models at LoD3 from LiDAR data and then loading these models into GE. Shortcomings of current techniques for creating virtual 3D cities that allow users to explore non-spatial business data of a building and its rooms is then discussed. Finally, a brief conclusion is given and direction for future work is presented.

## 2 Related Work

In 3D geo-information cities, building models linked to non-spatial data operate on web-based and wireless platforms where users can easily interact and explore building attribute information. In this city, 3D building models are core and can be created from several resource materials involving LiDAR (aerial, terrestrial and mobile) scanning data, aerial images, photographs and architectural drawings. As such, this section provides an overview of existing work on building reconstruction from LiDAR data and more recent work on virtual 3D city modeling.

### 2.1 3D Building Reconstruction

To create 3D building models, several approaches have been reported that semi-automatically [3] or automatically [4, 5] reconstruct 3D building models from LiDAR data (both airborne laser scan (ALS) and terrestrial laser scan (TLS)) sometimes combined with photogrammetry or 2D building "footprints". The processes can be divided into either *model-driven* or *data-driven* approaches.

In the former, geometric primitives (solids) are initially described and building features (walls, roofs) are subsequently fitted to point cloud data [6, 7]. For example, Haala et al. [6] proposed four different roof primitives and their combinations to derive automatically the 3D building geometry of houses from ALS and existing ground planes. Similarly, Maas and Vosselman [8] introduced an invariant moment based algorithm for parameters of a standard, gabled-roof house type, which allowed for modeling asymmetric elements such as dormers. However, these efforts assume homogenous LiDAR point distributions, which is unrealistic. You et al. [9] also adapted a set of geometric primitives and fitting strategies to model complex buildings with irregular shapes, but the approach required user intervention and generated only limited wall details. Hu et al. [7] also used a combination of linear and non-linear fitting primitives to reconstruct a complex building, in which aerial imagery was used to refine the models. In summary, this technique has limited applicability to complex buildings and results in relatively simple representations of 3D building models.

With data-driven techniques, building boundaries and features are generated directly from ALS data [5, 10, 11] and from TLS data [3, 12-14]. When using ALS data, the technique reconstructs roof shapes directly from sample points of roof planes. Subsequently, the remainder of the building is simply extruded down to the ground level from the roof shape outlines. Vosselman and Dijkman [11] used a Hough transform for extraction of plane faces (roof planes) from the ALS data, and then 3D building models were reconstructed by combining ground planes and the detected roof planes. Similarly, Henn et al. [15] used RANSAC/MSAC (RANdom SAmple Consensusr/M-Estimator Sample Consensus) to produce roof models provided by the international standard CityGML from ALS data. The vertical wall was generated by extruding up from 2D building footprints to obtain a complete

building model with LoD2. Hofmann et al. [16] introduced a method to extract planar roof faces by analyzing triangle mesh slopes and orientations from a Triangular Irregular Network (TIN) structure generated from ALS data. More recently, Dorninger and Pfeifer [10] used an  $\alpha$ -shape approach to determine a roof outline from point clouds of the roof projected onto a horizontal plane. Also, Zhou and Neumann [5] created impressive buildings for a large urban area by using a volumetric modeling approach, in which roof planes were determined based on a normal vector obtained from an analysis of grid cells belonging to roof layers. Such methods can be applied to reconstruct arbitrary shapes [3, 17]; however, this approach is more sensitive to noise from the input ALS data than model-driven approaches and as building models are extruded, they continue to lack any vertical wall (building facade) details.

When data-driven techniques are applied to TLS data, points lying on a dataset's boundaries can be extracted from input data using angle criterion [13] or Delaunay triangulation [3, 12, 14]. Outlines of buildings and their facade features (e.g. doors and windows) are subsequently generated from those boundary points. The completed models are achieved by mapping selected images (photos) to building model surfaces [3]. This work was successful in reconstructing a simple, mostly flat, facade containing simple morphology.

Interactive tools (some commercially available [18]) have also been developed to support users in the fast reconstruction of building models [19]. Despite these significant advances, building models reconstructed from ALS data still just represent a building's outline and its roof, and still require further manual post-processing to generate more geometric detail. In addition, models generated from TLS data can describe details of vertical surfaces of a building, but then is still missing the roofs as insufficient data is collected. Thus, the efficient construction of 3D building models that contain a high level-of-detail (LoD3/4) from LiDAR point data is still an open problem.

## 2.2 3D City Modeling

In constructing a detailed 3D city model, a lot of spatial and non-spatial information is necessarily gathered and linked to complete an object's description. In this regard, 3D building models that integrate business data are stored in a DBMS from which they can be efficiently exported for GIS or web-based visualization [20, 21]. Kolbe et al. [22] decomposed a 3D city model into CityGML, which covers the geometrical, topological, and semantic aspects of spatial objects. Additionally, a 3D spatial database for CityGML was proposed to manage large CityGML datasets by optimizing the database schema and data access tools [23]. In this work, an import tool loads a CityGML dataset into an Oracle DBMS while an export tool allows for querying the database according to user-defined criteria, and subsequently transforms thematic information and structures to 3D graphics formats like KML and X3D/VRXL. In developing their web-based 3D campus information system,

Kahraman et al. [24] manually created LoD2 building models with Sketchup software based on CAD files and textured photos of the facade. Subsequently, the textured 3D buildings were exported into Graph Modelling Language (GML) format for storage and visualization. A web-based viewer called the WebViewer was developed to allow the possibility for exploring and visualizing the building models in the system.

### 3 Generating 3D Building Models

As building models with LoD2 were previously used for 3D city modeling, they did not integrate non-spatial data of sub-objects (e.g. rooms) in the building model. To provide sufficient details of building models for integrating non-spatial data, a new workflow is proposed to reconstruct LoD3 building models from TLS data with full detail of exterior walls and roofs included (Fig. 1). In this work, point clouds of buildings were first acquired using Leica ScanStation C10, which was controlled by the proprietary software Cylone installed on a laptop linked to the scanner [25]. Subsequently, the point clouds were registered and geo-referenced within the Cylone environment and the 3D building models were manually created using AutoDesk and a CloudWorx plug-in. CloudWorx offers many manipulation and editing tools to assist users to trace or auto fit lines, arcs and polylines to 3D point cloud data [26]. Finally, by employing FEM Workbench [27], the 3D building model's underlying CAD geometry is transformed to KML format to allow for display in web-based mapping applications in Google Earth.

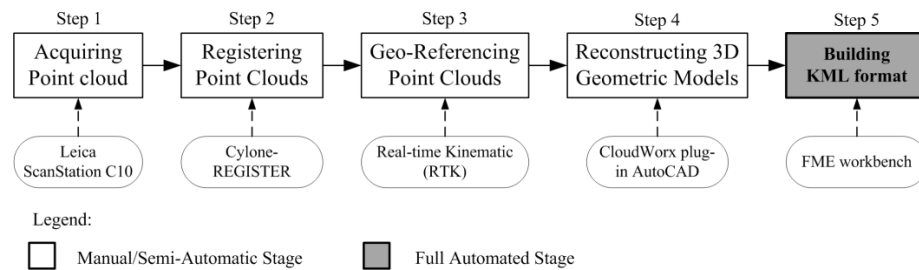


Fig. 1. Workflow of building reconstruction

The methodology was applied to the north campus of the National University of Ireland in Maynooth (NUIM) as the study area (Fig. 2). This was selected because the area contains a mixture of simple and complex buildings with various architectural styles (e.g. historic and modern buildings), which can be most problematic when reconstructing 3D building models. The highest building is 20 m.



**Fig. 2.** North campus study area of NUIM. Modeled buildings inside dashed polygon.

In step 1, 14 buildings in the NUIM study area were scanned using a Leica ScanStation C10. Due to the complexity of some buildings, multiple scan stations were set up around a building to ensure sufficient point cloud coverage (Table 1). Each dataset was stored in a local coordinate system centered around the location of the scanner. A resolution of 10 mm point spacing was set for the data collection process. In fact, as north campus buildings are distributed sparsely, this allowed for TLS to acquire sufficient point clouds of some roofs as well.

To prepare the point clouds for building reconstruction, the datasets from each scanning station were registered and merged into one dataset stored in a single coordinate system using the Cylone-register module. In this process (step 2), a source dataset is manually merged to a target dataset by mapping a pair of reference points from the target and source datasets. There are three reference points for each dataset, and are often selected from sharp angular features on the building facade, for example window corners or lintel edges. The registration process is completed when the average distance between two data points in an overlapping region is less than a threshold that can reduce noisy dataset, for example, Truong-Hong [28] uses a threshold of 5mm.

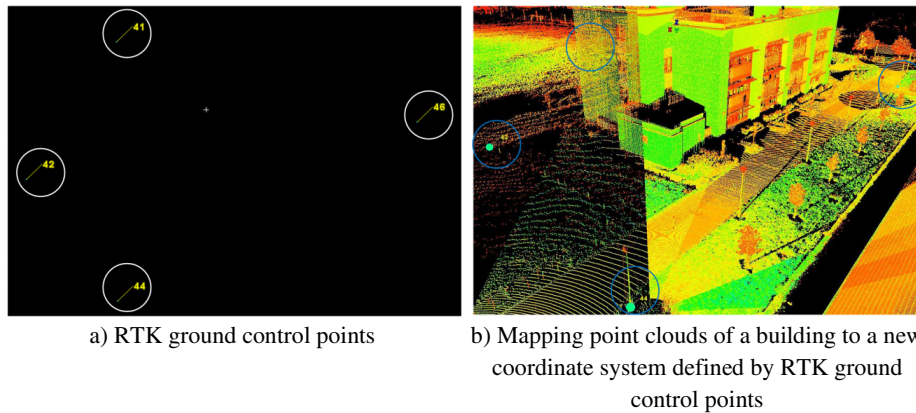
Subsequently, the Irish National Grid coordinate system was assigned to the point cloud in the final dataset to geo-reference the entire campus model allowing for geographically accurate visualization applications (step 3). Real Time Kinematic (RTK) positioning was used to get accurate X, Y, and Z ground control points. Approximately, 50 observations were taken in total in the North campus with a maximum elevation error less than 1cm. To obtain control points at each chosen location, residence times varied from 2-4 minutes for each point. After importing



RTK ground control points into the Cylone program, common data points from the LiDAR building dataset were mapped to the RTK ground control points within the ScanWorld module of the Cylone program (Fig. 3).

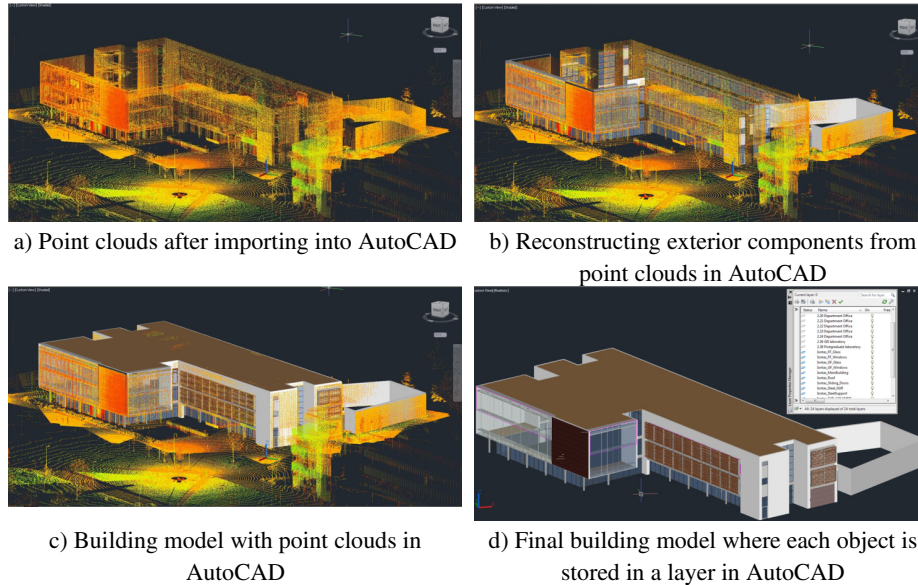
**Table 1.** Configuration of datasets

No.	Building name	No. scan stations	No. points (million)
1	Arts block	8	16.19
2	John Hume	16	5.77
3	Science	12	13.49
4	Callan	18	20.31
5	Engineering	15	15.45
6	Canteen and sport center	9	15.38
7	Hamilton and Rye Hall	36	44.87
8	Student Union	6	7.41
9	Iontas	12	6.00
10	Education House	18	18.86
11	St. Anne's	12	14.55
12	St. Catherine's	11	6.60
13	Student Services	13	13.32
14	Auxilia	17	8.09



**Fig. 3.** Process of geo-referencing point clouds

After registering and geo-referencing the building point clouds, they were imported into the AutoCAD environment through a Leica plug-in called CloudWorx, which is a tool for viewing and working with slices of point cloud data to reconstruct 3D solid models directly from LiDAR point clouds [26]. For example, Fig. 4 illustrates the process of building reconstruction (step 4). During reconstruction, details of the building, e.g. windows/door frames or balconies and photographs of the building are used as reference.



**Fig. 4.** Process of Iontas building reconstruction from LiDAR point cloud data

Non-spatial business data is integrated with spatial data in order to better describe a building's various functions and purpose. However, the solid building model created in the previous step has limitations concerning access to its attributes in Google Earth. A restriction of GE when displaying/querying 3D solids is that linked attribute information cannot be accessed by pointing (clicking/selecting) anywhere on the solid shape. To overcome this, the solid building shape must be converted to 3D polygon data, as GE does recognize clicking anywhere inside a polygon shape to select/query the object. However, automatically converting 3D solid window/door shapes to 3D polygons results in a minimum of 16 points per window/door object, and with many windows/doors associated to each campus building, these many points soon overwhelm the real-time display capabilities of GE. The solution to this GE display problem was to leave the windows/doors in solid format for realistic 3D display purposes and manually add an overlay of 3D polylines around window/door borders within the CAD drawing, where the polyline outline only consists of 4 points per object.

Both 3D solids and 3D polylines are managed in separate layers in the AutoCAD database. There are two groups of layers: (i) stored solid components involving exterior vertical walls, window/door frames, roofs, and balconies and (ii) stored polygons of window/door extents. The layer name is assigned based on its group. For example, with the Iontas building model in Fig. 4, if the layer belongs to group (i), its name incorporates the building name plus the component name (e.g. a main wall is stored in the layer named "*Iontas\_MainWall*" while a window frame is stored in the layer named "*Iontas\_WindowFrames*"). Similarly, if the layer is in group (ii), the layer name describes the number and name of the room. For example, in the Iontas building, the window polylines of room 2.36 GIS Laboratory is stored in the layer

named "2.36 GIS Laboratory". This data management approach allows for efficient geometry extraction of building components in the next step.

## 4 Virtual 3D Building Models

There are several commercial and free mapping products available that allow users to incorporate 3D building models for visualization and interaction applications. Of these, Google Earth was selected for displaying the 3D buildings used in this study because it is free and widely popular with both web and wireless GIS users. GE allows users to upload 3D building models for direct visualization in the GE environment or for inclusion in a webpage using their application programming interface (API). When retrieving data for visualization, a client (desktop or mobile) queries for data either from temporary cache memory or from GE databases. In order to include 3D building models in GE, the original models stored in CAD format and associated to non-spatial attribute data are transformed into the KML format using the FME Workbench utility [27].

Our aim is to place detailed 3D building models within the popular GE online web mapping environment where the general public can freely access and interact with the models to query and retrieve information about building details, uses, and contents (step 5). For example, users can click on exterior walls or roofs of the building to query the building's name and its purpose. Also more specific queries can be asked by clicking on individual building features like windows or doors to retrieve full details about business data concerning the function of a building's individual rooms.

The building models created in the previous section involve 3D solids and polylines that accurately represent the 3D geometries of a building and any visible interior rooms. The KML format supports both solid and polygon types where 3D solids are stored in the COLLADE interchange file format. However, GE does not allow users to query for non-spatial data of a building simply by clicking on the building components when a COLLADE file is used to describe building solids. Thus, we decomposed the 3D solid building geometries into KML format as 3D polygons, and also applied this data type to the 3D polylines representing the spatial coordinates of its rooms. A workflow was developed to transform building model CAD files to KML through two schemas: (i) for geometries of the building and (ii) for geometries representing rooms (Fig. 5).

The geometries of building components (e.g. main walls, roofs, balcony or window/doors) are managed by two groups of layers in an AutoCAD database. A "StringSearching" transformer [box (a) in Fig. 5] was used to separate geometries of the building and rooms based on the AutoCAD layer names. Subsequently, a series of "StringSearching" transformers [box (b) in Fig. 5] continuously extracted geometries of each component of the building model. 3D solids of the building components in a CAD file are now transferred into KML file as polygons [box(c) in Fig. 5] and solids [box (d) in Fig. 5] using the "GeometryCoercer" transformer. Of which, 3D solids are only used for geometries of window/door frames while polygons are applied to all

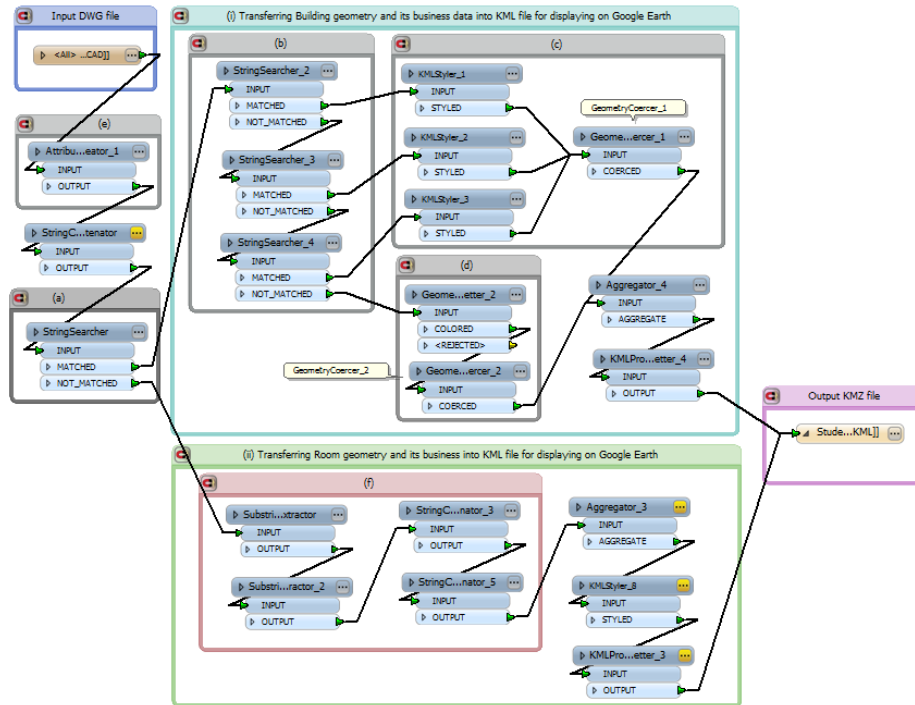


Fig. 5. A workflow to transform 3D building models from DWG to KML

other objects. This step allows us to assign an appearance to each building component by filling with either a colour or texture. In this study, the appearance of a building is defined by a filling colour through the "*KMLStyle*" transformer [box(c) in Fig. 5] for polygons and "*GeometryCoercer*" transformer [box(d) in Fig. 5] for solids. Additionally, the geometries of polyline rooms automatically get transformed to 3D polygons in the KML file.

Any non-spatial attribute data associated to a building and its rooms can be manually assigned using the "*AttributeCreator*" transformer (box (e) is Fig. 5). Attributes such as address, purpose, height, date of construction, and total building volume were assigned to each 3D building model. A series of attributes involving capacity, facilities, and schedules were entered for each room. Additionally, a room's number and name was established using the "*SubstringExtractor*" and "*StringConcatenator*" transformers (box (f) in Fig. 5). A URL link to a webpage can be added through the "*AttributeCreator*" transformer to allow for dynamic retrieval of business data via external web content. Consequently, the final KML file of the fully attributed building model is now ready for access in GE, as shown in Fig. 6. In this environment, users can now query for non-spatial data of a building and rooms separately by clicking on exterior building components (Fig. 6a) or sub-components like windows (Fig. 6b), respectively. The resulting building



a) Retrieving non-spatial data of a building



b) Retrieving spatial business data of a room

**Fig. 6.** Visualization and interaction with John Hume building in GE

models for the entire study area as displayed in GE is shown in Fig. 7, where every component and sub-component of every building is individually available for contextual querying.



Fig. 7. Visualization of detailed 3D models of NUIM north campus in GE

## 5 Discussion

Through this demonstration of converting LiDAR point clouds to geometries of building models and transforming them to KML format for interaction and visualization in Google Earth, several limiting factors emerged. When creating 3D virtual cities for *Internet of Things* type applications and general mobile spatial interaction, the most important task is to generate detailed and geometrically accurate building models. In contrast to traditional methods (e.g. on-site surveying), TLS is emerging as an attractive alternative for collecting building coordinate data in terms of field time and accuracy [12, 13, 28]. For example, it took around 80 minutes to collect the necessary point clouds of the Iontas building (Fig. 4), in which the dataset contained approximate six millions of points. Scanning time can be further reduced approximately by half if the scan resolution doubles to 20 mm [29]. However, the process of building 3D models from this data is more time consuming, and takes around 6-8 hours for each building. For example, it took 7 hours to reconstruct the Iontas building model (Fig. 4). This implies automatic or semi-automatic processes must be developed to reconstruct building models with LoD3 to reduce post-processing bottlenecks for city-wide 3D modeling workflows.

Interestingly, as our study area is mostly rural where buildings are low and sparsely distributed, TLS can sufficiently acquire point clouds of not only building facades but also roofs. This advantage allows for building models to be reconstructed from point cloud resources only. However, for urban areas, TLS cannot often collect point clouds of roofs. In this case, ALS or photogrammetric data as an additional resource for

generating roof features is needed [30]. By using RKT ground control points to georeference the point clouds, the geographic location of the finished models is highly accurate which enables accurate mobile spatial interaction applications.

Currently, 3D building models with LoD2 can be automatically generated from LiDAR data [31]. However, this category of modeling has limited non-spatial attribute potential for describing sub-components (like rooms) if these models are destined for exploration in online web mapping environments like Google Earth. The workflow proposed in this paper was successful in reconstructing geometrically accurate building models with LoD3. However, the procedure is time consuming for larger project areas where numerous building models need to be reconstructed; therefore, automation of this approach is still an open problem and left for future work. Additionally, the process itself fails to create some intricate details of facade architectures, so supporting imagery (e.g. Google StreetView) may be required to obtain ever more photo realistic models. As noisy point clouds are common, additional reference photos of the building may be needed for reconstructing some windows/door details.

To allow users to query the non-spatial attributes of 3D objects in GE simply by clicking on the object, all geometries of the building models were necessarily converted to KML polygon format, where the model then gets rendered based on a triangular mesh. However, given the complexity of 3D building models, especially when all building sub-components are also included, the number of polygon vertices may exceed the limits supported by GE for real-time display over the web. This leads to parts of the model or models not rendering simultaneously. To overcome this shortcoming, the complexity of the model in the KML file must be reduced. One solution is to first import the building CAD file into Google Sketchup to reduce the complexity and detail of the model and then texture the model before exporting to the KML file. While this method improves the visual reality of the building model, non-spatial business data cannot be linked through Google Sketchup. Another solution proposed by this study is to not transform all model geometries directly to KML polygon format. Instead, the KML file maintains links to solid window/door frame objects stored in a COLLADE interchange file and displays them directly as such.

In this way, users can click on both 3D polygons representing a building's structural components (e.g. facade or roof) when querying for non-spatial attribute data and can also click on windows and doors to retrieve any business data related to interior rooms. This is accomplished using an overlay of 3D polylines to represent the window/door borders within the CAD drawing before transforming to KML format. Additionally, a further reduction of the total number of points in the KML file can be done by transforming to 3D solids any other small building details (e.g. architectural elements like gables, etc.) that do not have any associated business data.

The above workflow provides a framework for bringing detailed LoD3 building models into GE where users can easily query for non-spatial data down to room level. Based on this workflow, individual or collections of models can be built and placed online in GE for real-time visualization and interaction within downstream value-added applications. As the GE platform currently only displays 3D objects above the earth's surface, this restricts visualisation of any underground detail below ground

elevation level. To display and interact with entire building models above and below ground, an alternate web mapping platform should be considered. Additionally, when the Google Earth interface is integrated with *RESTful (Representational State Transfer)* web-services, it allows for displaying the building model at various levels of detail. Indeed, users can discover different information for a building depending on their distance from it, as described in Pham-thi et al. [20]. In the case of displaying buildings with LoD4, structural components associated with non-spatial data for each story can be stored in each KML file separately, and *RESTful* web-services used to activate/de-activate each story to allow users to explore any available interior attributions of the building.

## 6 Conclusions and Future Work

The development of detailed 3D cities allows users to visualize and interact with building models that can act as an interface for accessing their business data. However, querying business data at room level is still not common in today's online web mapping platforms because contemporary models are LoD2, with no attribute details at room level. This paper proposed a workflow to construct building models with LoD3 from TLS data and business data for both the building and its rooms and subsequently upload the model for real-time display in GE. The reconstructed building model has sufficient geometric detail to allow us to define business data for all visible rooms from the outside of a building. The manual building reconstruction process showed to be time consuming and dependant on developer experience.

In the next step, a (semi-)automatic approach will be developed to addresses some of the workflow bottlenecks when reconstructing building models - for example when converting 3D solids to simplified 3D polygons. This new work will focus on reducing the post-processing time. However, because of the complexity of real-world buildings, this is still a big challenge. In addition, as the version of GE used in this study was free, some limitations on KML support for displaying high complexity models and when displaying parts of the model below the earth's surface were encountered. As such, other web-mapping platforms (e.g. Bing, OSM, NASA) should be looked at to potentially address this restriction.

**Acknowledgements.** Research presented in this paper was funded by a Strategic Research Cluster Grant (07/SRC/I1168) by Science Foundation Ireland under the National Development Plan. The authors gratefully acknowledge this support. We also thank SAFE software for providing a research grant of FME workbench. Thanks to Damien Butler of National Centre for Geocomputation, NUIM for providing data and modeling expertise.

## References

1. Haala, N., Kada, M.: An Update on Automatic 3D Building Reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing* 65, 570–580 (2010)



2. Kolbe, T.H., Nagel, C., Stadler, A.: CityGML-OGC standard for photogrammetry? *Photogrammetric Week*, 265–277 (2009)
3. Pu, S., Vosselman, G.: Knowledge based reconstruction of building models from terrestrial laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing* 64, 575–584 (2009)
4. Becker, S., Haala, N.: Refinement of Building Facades by Integrated Processing of LIDAR and Image Data. In: *Photogrammetric Image Analysis, PIA 2007*, Munich, Germany, September 19-21, vol. 36(33/W49A), pp. 37–12 (2007)
5. Zhou, Q.-Y., Neumann, U.: 2.5D Dual Contouring: A Robust Approach to Creating Building Models from Aerial LiDAR Point Clouds. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) *ECCV 2010, Part VI. LNCS*, vol. 6316, pp. 1–14. Springer, Heidelberg (2010)
6. Haala, N., Brenner, C., Anders, K.H.: 3D Urban GIS From Laser Altimeter And 2D Map Data. *International Archives of Photogrammetry & Remote Sensing* 32, 339–346 (1998)
7. Hu, J., You, S., Neumann, U., Park, K.K.: Building Modeling from LiDAR and Aerial Imagery. In: *ASPRS 2004*, pp. 1–6 (2004)
8. Maas, H.-G., Vosselman, G.: Two Algorithms for Extracting Building Models from Raw Laser Altimetry Data. *ISPRS Journal of Photogrammetry & Remote Sensing* 54, 153–163 (1999)
9. You, S., Hu, J., Neumann, U., Fox, P.: Urban Site Modeling from LiDAR. In: *Proceedings of the 2003 International Conference on Computational Science and its Applications*, pp. 579–588 (2003)
10. Dorninger, P., Pfeifer, N.: A Comprehensive Automated 3D Approach for Building Extraction, Reconstruction, and Regularization from Airborne Laser Scanning Point Clouds. *Sensors* 8, 7323–7343 (2008)
11. Vosselman, G., Dijkman, S.: 3D Building Model Reconstruction from Point Clouds and Ground Planes. In: *International Archives of Photogrammetry and Remote Sensing*, pp. 37–43 (2001)
12. Truong-Hong, L., Laefer, D.F., Hinks, T., Carr, H.: Flying Voxel Method with Delaunay Triangulation Criterion for Façade/Feature Detection for Computation. *ASCE Journal of Computing in Civil Engineering* 26, 691–707 (2012)
13. Truong-Hong, L., Laefer, D.F., Hinks, T., Carr, H.: Combining an Angle Criterion with Voxelization and the Flying Voxel Method in Reconstructing Building Models from LiDAR Data. *Computer-Aided Civil and Infrastructure Engineering* (2012), doi:10.1111/j.1467-8667.2012.00761.x
14. Boulaassal, H., Landes, T., Grussenmeyer, P.: Automatic extraction of planar clusters and their contours on building façades recorded by terrestrial laser scanner. *International Journal of Architectural Computing* 7, 1–20 (2009)
15. Henn, A., Gröger, G., Stroh, V., Plümer, L.: Model driven reconstruction of roofs from sparse LIDAR point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing* 76 (2013)
16. Hofmann, R.D., Maas, H.-G., Streilein, A.: Derivation of Roof Types by Cluster Analysis in Parameter Spaces of Airborne Laserscanner Point Clouds. In: *ISPRS Commission III WG3, Workshop, 3-D Reconstruction from Airborne Laserscanner and InSAR Data. IAPRS International Archives of Photogrammetry and Remote Sensing and Spatial Information Sciences* (2003)
17. Chen, J., Chen, B.: Architectural Modeling from Sparsely Scanned Range Data. *Journal International Journal of Computer Vision* 78, 223–236 (2008)
18. Kubit: PointCloud (1999)

19. Nan, L., Sharf, A., Zhang, H., Cohen-Or, D., Chen, B.: SmartBoxes for interactive urban reconstruction. *Journal ACM Transactions on Graphics* 29, Article 93 (2010)
20. Pham Thi, T.T., Truong-Hong, L., Yin, J., Carswell, J.D.: Exploring Spatial Business Data: a ROA Based eCampus Application. In: Liang, S.H.L., Wang, X., Claramunt, C. (eds.) *W2GIS 2013. LNCS*, vol. 7820, pp. 164–179. Springer, Heidelberg (2013)
21. Wu, H., He, Z., Gong, J.: A virtual globe-based 3D visualization and interactive framework for public participation in urban planning processes. *Computers, Environment and Urban Systems* 34, 291–298 (2010)
22. Kolbe, T.H., Gröger, G., Plümer, L.: CityGML – Interoperable Access to 3D City Models. In: *International Symposium on Geo-Information for Disaster Management*. Springer (2005)
23. Stadler, A., Nagel, C., König, G., Kolbe, T.: Making Interoperability Persistent: A 3D Geo Database Based on CityGML. In: Lee, J., Zlatanova, S. (eds.) *3D Geo-Information Sciences*, pp. 175–192. Springer, Heidelberg (2009)
24. Kahraman, I., Karas, I.R., Rahman, A.A.: Developing web-based 3D campus information system. In: *10th International Symposium & Exhibition on Geoinformation (ISG2011) & ISPRS Commission II/5 & II/7 Conference*, p. 6 (2011)
25. Leica Geosystems AG: *Leica Cyclone-3D Point Cloud Processing Software* (2011)
26. Leica Geosystems AG: *Leica CloudWorx for AutoCAD* (2011)
27. Safe software: *FME desktop* (2013)
28. Truong-Hong, L.: *Automatic Generation of Solid Models of Building Façades from LiDAR Data for Computational Modelling*. School of Architecture, Landscape and Civil Engineering. PhD. University College Dublin (2011)
29. Truong-Hong, L., Laefer, D.F.: Validating Computational Models from Remote Sensing Data for Historic Facades. *ASTM, Journal of Testing and Evaluation* (accepted 2013)
30. Laefer, D.F., Hinks, T., Carr, H., Truong-Hong, L.: New Advances in Automated Urban Model Population through Aerial Laser Scanning. *Recent Patents in Engineering* 5, 196–208 (2011)
31. Henn, A., Gröger, G., Stroh, V., Plümer, L.: Model driven reconstruction of roofs from sparse LIDAR point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing* 76, 17–29 (2013)