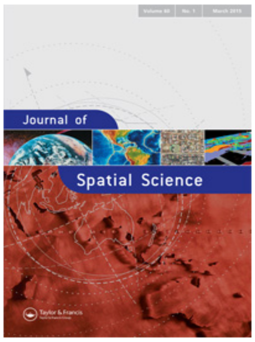


Taylor & Francis
Taylor & Francis Group

Journal of Spatial Science

ISSN: 1449-8596 (Print) 1836-5655 (Online) Journal homepage: http://www.tandfonline.com/loi/tjss20

# Planning and visualising 3D routes for indoor and outdoor spaces using CityEngine

KyoHyouk Kim & John P. Wilson

**To cite this article:** KyoHyouk Kim & John P. Wilson (2015) Planning and visualising 3D routes for indoor and outdoor spaces using CityEngine, Journal of Spatial Science, 60:1, 179-193, DOI: 10.1080/14498596.2014.911126

**To link to this article:** http://dx.doi.org/10.1080/14498596.2014.911126

Published online: 27 May 2014.

*Journal of Spatial Science*, 2015
Vol. 60, No. 1, 179–193, http://dx.doi.org/10.1080/14498596.2014.911126



**PROFESSIONAL PAPER**

# Planning and visualising 3D routes for indoor and outdoor spaces using CityEngine

KyoHyouk Kim* and John P. Wilson

*Spatial Sciences Institute, University of Southern California, Los Angeles, CA, USA*

With the increasing size and complexity of modern buildings, 3D indoor routing is receiving more attention nowadays. Different elements such as route finding, indoor modelling, and route visualisation need to work together to achieve this goal. For this purpose, we propose a framework that makes use of existing data sources and tools that can minimise the time and effort needed for potential 3D indoor routing applications. Our 3D indoor and building models are generated from CAD files and building footprints using CityEngine and its built-in procedural modelling approach. An Americans with Disabilities Act (ADA) compatible 3D network is created by combining 3D floor lines and transitions such as staircases and elevators. The resulting routes as well as the indoor and façade models are then visualised through a 3D WebScene generated by CityEngine. Our study demonstrates the usefulness of CityEngine for 3D indoor model generation as well as 3D routing visualisation.

**Keywords:** CityEngine; procedural modelling; CGA shape grammar; 3D-GIS; 3D indoor routing

## 1. Introduction

The three-dimensional (3D) building model is one of the essential components in a range of applications, such as urban design, city planning, virtual reality, entertainment, solar potential estimation, and emergency response. Numerous approaches have been proposed using different data sources, such as Digital Surface Models (DSMs) (Maas 1999; Vosselman 1999; Rottensteiner & Briese 2002), LiDAR point clouds (Sampath & Jie Shan 2010; Huang *et al.* 2011; Kim & Shan 2011) or airborne stereo imagery (Baillard & Maıtre 1999; Fraser *et al.* 2001). More details and comprehensive overviews of this topic are presented in Haala and Kada (2010). Apart from these efforts, 3D virtual scenes, including building models as well as other features of cities (e.g. roads, trees, etc.) have been generated by, and continue to rely on, either manual creation or semi-automatic approaches based on photogrammetry, computer vision, or generative modelling techniques (Müller Arisona *et al.* 2013). In this context, it has been possible to rapidly reconstruct 3D virtual scenes of high visual quality for the last few years, especially for major cities in many countries. These examples have led the way in helping us envision the need for more detailed models or new applications.

The typical Geographic Information Systems (GISs) that operated in 2D are now swiftly moving towards 3D space. In the same way, 3D building modelling is also undergoing

*Corresponding author. Email: kyohyouk@usc.edu

the same transition, such that the indoor spaces of buildings are now being described as well. Obviously, this trend is giving rise to new applications, such as the indoor modelling, routing, and visualisation tasks that are the focus of our work.

With the increasing size and complexity of modern buildings, indoor routing is becoming increasingly important. It is obvious that people who spend a great deal of time indoors are often confronted with foreign environments (Winter 2012). Visitors are very likely to have difficulties finding the optimal route to their destination within complex environments such as multi-level public buildings and university campuses. Furthermore, for people with special needs (e.g. the mobility-impaired), a conventional guidance system based on either a rough map or one or more accessibility sign(s) is not sufficient. To overcome these limitations, more advanced routing systems making use of state-of-the-art technology, such as 3D GIS and indoor positioning and routing systems, are needed. Moreover, indoor routing systems can be used for facility management and emergency response; for example, in the case of a critical incident, timely response and decision-making are necessary to provide evacuation routes that can minimise the loss of life and/or property (Liu *et al.* 2010). Recently, some large companies with geospatial portfolios, such as Google, Microsoft and Navteq, have begun providing indoor layout information as well as indoor routing. However, it is currently limited to 2D visualisation, such that only one level can be depicted (Goetz 2013).

To achieve a fully operational indoor routing application, different elements, such as an indoor model, route finding, and route visualisation, should work together (Figure 1). Considering 3D visualisation and the accompanying web-based services, additional factors, such as compatibility and performance, should also be taken into consideration. Many studies have been carried out on this topic, most of which are primarily focused on specific elements (i.e. designing 3D network models, proposing routing algorithms (Kwan & Lee 2005; Meijers *et al.* 2005; Dudas *et al.* 2009; Karimi & Ghafourian 2010; Li *et al.* 2010), and generating and visualising 3D

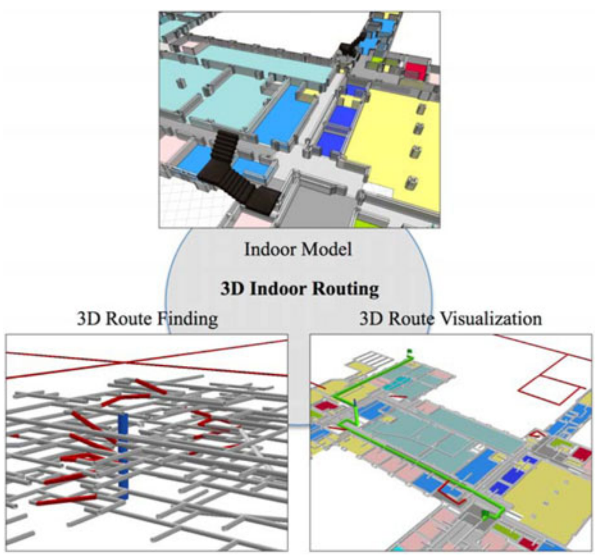


**Figure 1.** Core elements for 3D indoor routing applications

indoor models (Biber *et al.* 2004; Horna *et al.* 2007)). However, less attention has been given to the framework for making use of existing resources to support 3D indoor routing and its visualisation. Utilising existing software and combining relevant functionalities can substantially reduce the time and effort needed to achieve this goal. Given this context, this article shows how existing tools can be combined with relevant functionalities to generate 3D models, especially for indoor spaces, to determine the best 3D indoor route(s), and to visualise the resulting 3D information.

The remainder of this article is structured as follows. Section 2 briefly introduces CityEngine and its built-in modelling language used to generate and visualise indoor and façade models. In Section 3, the workflow of our study is presented in detail. Thereafter, Section 4 uses an example building on the University of Southern California (USC) campus to show the routing results as well as the visualisation. We summarise and offer final remarks and thoughts for future work in Section 5.

## 2. CityEngine and CGA shape grammar

CityEngine is a 3D modelling application used for generating large-scale buildings and virtual cities. It was originally developed by Procedural Inc., and is now being further developed by the Esri Zurich R&D Center. The CityEngine application relies on procedural modelling (Parish & Müller 2001; Müller *et al.* 2006), which in turn was based on L-Systems (Prusinkiewicz & Lindenmayer 1990). CityEngine supports a built-in programming language called 'Computer Generated Architectures (CGA) shape grammar' for procedural modelling that was originally developed by Müller *et al.* (2006).

CGA shape grammar is an extension of the set grammars introduced by Wonka *et al.* (2003). The improved capabilities include tools for the modelling of architecture (Halatsch *et al.* 2008). CGA shape grammar applies numerous sets of shape grammars, such as translation, scale, rotate, and split, to the initial shape, such as building footprints in 2D or shape volumes in 3D. In CityEngine, this set of operations is organised sequentially, thereby forming a CGA rule file. Users can make their own CGA rules using the built-in editor in CityEngine. For the initial shape, CityEngine can import different types of geospatial datasets and refer to the associated semantic information, which then can be selectively used for generating unique 3D models for different scenarios. One of the advantages of CGA shape grammar over the preceding shape grammars is that it supports component-splitting to allow shapes to be divided into shapes of lesser dimensions (Müller *et al.* 2006). For instance, once the CGA shape grammar is used to create an original mass model from the initial shape, it can then be used to model the façade and add more details for windows, doors and ornaments (Müller *et al.* 2006). This sequential application of rules is effective for describing the spatial distribution of features and components for architectural shapes (Prusinkiewicz *et al.* 2001).

As an example, Figure 2 shows how CGA shape grammar can be used for building façade modelling. The original mass model is first generated from the initial footprint (Figure 2a). The first façade is then subdivided into one ground floor and upper floors. Subsequently, each floor is further split into a bottom ledge, tile (i.e. window), and a top ledge area, depending on each floor index. Each tile is further divided into the window block composed of the centre window and wall areas on both ends, which is repeated for the predefined number of tiles and window blocks. Figure 2b illustrates the façade model produced using this procedure. A more detailed façade can also be modelled by inserting predefined architectural shapes or applying corresponding textures as shown in Figures 2c and 2d.

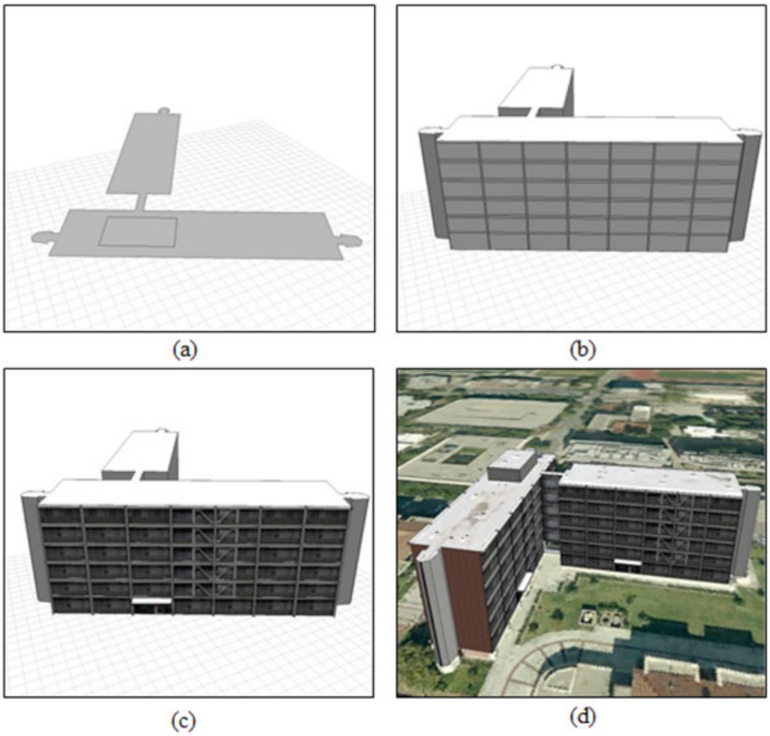


**Figure 2.** (a–c) Three examples of building façade models generated with CGA shape grammar; and (d) a CGA-generated 3D model of the Parkside Apartments at USC

## 3. Workflow

This section details the work tasks we performed in this study. As shown in Figure 3, we first generated 3D indoor and building façade models using CityEngine from CAD files and building footprints, respectively. A 3D network dataset was created, from which 3D routing was performed in ArcScene. The resulting routes, as well as the 3D models, were then published to a 3D web scene using CityEngine for visualisation, which can be made accessible using Internet browsers supporting WebGL. The details of each step are discussed in the following subsections.

### *3D Modelling indoor spaces*

#### *Extraction of principal features*

A 3D indoor model is primarily used for visualisation in 3D indoor routing applications. In this context, varying Level-of-Details (LoDs) can be achieved, depending on the specific needs of the users or the applications as shown in Figure 4. Considering both visualisation performance and the time needed to build 3D models, we aimed to build 3D indoor models at LoD2.5 complexity. Consequently, the proposed indoor model includes various spaces such as offices, lobbies, hallways, and corridors, as well as interior walls; however, it does not show the shapes of doors and windows.

The principal features required for the indoor routing application were extracted from the CAD files by applying the geoprocessing model. The strategies were to identify the layer names associated with the principal features and extract them from the CAD layers. The geometry of the principal features was extracted from the polyline or polygon

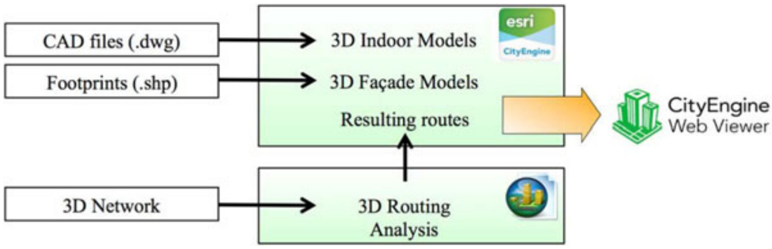


**Figure 3.** Overall study workflow

features, while the semantic information was determined from the point layer in the CAD file and added to the corresponding features and spaces through a spatial join. Due to some consistency issues inherent in the original CAD files, some manual processing was inevitable. Figures 5a and 5b show the original CAD file and the extracted polygons representing all the features and spaces, respectively. The semantic information for each feature or space was stored in the attribute table. Additional information that was not included in the input CAD files, such as the department name and resident's information, was compiled in a separate table and added to the polygon layer. The same strategy was used to extract the wall polygons, as shown in Figure 5c. Table 1 summarises the attribute fields included in the resulting polygon layer. The upper seven fields were determined from the input CAD file, and the others were joined from the separate table.

### *Generation of 3D indoor models*

Once we had extracted all the polygonal spaces, they were imported into CityEngine and the CGA rules were applied, thereby generating 3D indoor models. The CGA rules first assigned a different colour to each space, depending on its usage type, referred from the field *ROOM_DESCP* in the shape file and positioned the floor plan based on the floor level (i.e. field *FLOOR*). In the same way, another CGA rule was applied to the wall polygons. In this case, the CGA rule first positioned the wall polygons up to the same level as the floor plan and extruded the walls up to the predefined height. Combining these two models, we obtained a 3D indoor model for each floor. Figure 6 shows the resulting 3D indoor model of the basement in the Allan Hancock Foundation (AHF) building.

Since CAD files contain data in 2D space and miss some of the information about the connections between different floors, some manual processing is required, especially for staircases and elevators (Liu *et al.* 2010). In the original CAD file, an elevator was defined as a closed polygon with a unique layer name such that they could be extracted easily. However, a staircase was represented as a series of polylines as shown in Figure 7a and these features could not be defined as polygons using a simple geoprocessing model. Therefore, a series of connected polygons were drawn manually, and attribute fields, such as

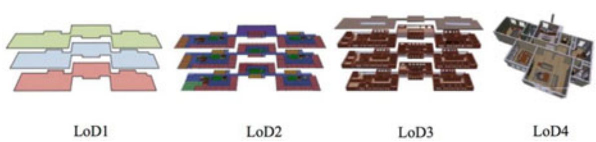


**Figure 4.** Different LoD of 3D indoor models (Hagedorn *et al.* 2009)

**Table 1.** Attribute fields for the resulting polygon layer

| Field Name | Data Type |
|---|---|
| FID | Object ID |
| Shape | Geometry |
| RefName | Text |
| RMNUMB | Text |
| RMDOOR | Text |
| ROOM_DESCP | Text |
| NET_SQFT | Double |
| DEPT_CODE | Text |
| SAC_CODE | Text |
| TITLE_CODE | Double |
| EMPLOYEE | Text |
| POSITION | Text |
| PHONE | Text |
| EMAIL | Text |
| FLOOR | Double |

*FLOOR_HT*, *S_FLOOR* and *E_FLOOR*, were prepared in ArcGIS. Then, a CGA rule was used to generate a 3D staircase model based on the starting floor level (*S_FLOOR*) and the vertical offset of each stair (*FLOOR_HT*). The resulting 3D staircase model in CityEngine is reproduced in Figure 7b.

### *3D network and route finding*

#### *Construction of 3D network*

Apart from the 3D indoor model, a 3D network dataset was necessary for performing the 3D indoor routing. For each floor plan, 2D floor lines were created by connecting all spaces to corridors or hallways as shown in Figure 8a. The associated attribute table included *FLOOR*, *ADA_RESTRICTED*, *PREFERENCE*, and *REAL_HGT* fields. The field *ADA_RESTRICTED* represents whether the given network segment restricts wheelchair access. Therefore, it was set to 0 for all ordinary paths, ramps, and curb ramps as well as elevators and set to 1 for staircases. The field *PREFERENCE* denotes the degree of preference of the given network segment to be used for determining an optimal route. A higher value (i.e. close to 1) was assigned to network segments passing through hallways or corridors, while a lower value was assigned to those that pass through offices. This approach effectively avoided finding routes that passed through a number of offices, as illustrated in Figure 9. The field *REAL_HGT* denotes the actual elevation above the ground of the given floor level, which was determined from the underlying DEM and the predefined floor height. After we built the 2D topology, the floor line features were converted to 3D line features using this *REAL_HGT* field.

In the same way, the transitions connecting different floors were also digitised first in 2D. The red line segments in Figure 8a show the centre lines of the four staircases connecting the basement and the ground floor. The start and end points of each transition should be connected to the end points of upper and lower floors. The associated attribute table incorporates additional fields such as *T_FLOOR* and *T_ELEV*, which denote the floor number and actual elevation, respectively, of the connected floor. Finally, the existing road network adjacent to the building was also connected to the entrances to the building. Figure 8b shows the completed 3D multimodal network composed of the campus road (green), the indoor floor lines (grey), the staircases (red) and the elevator (blue). Using all these feature datasets, we created a 3D network dataset in ArcGIS Network Analyst.

#### *Best route finding*

Esri's ArcGIS Network Analyst supports various network-based analysis tasks, including the calculation of routes, closest facilities, and service areas. For the routing analysis, numerous network parameters can be tuned during the network-building process. Obviously, a more advanced and detailed route-finding task can be performed if the input feature datasets include all the relevant information for both the geometry and the attributes. While conventional outdoor routing is based on criteria such as fastest, shortest and least turns, indoor routing is primarily based

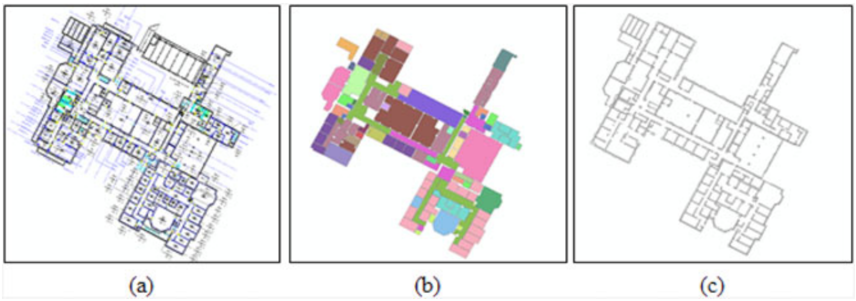


**Figure 5.** Extraction of principal features, with sketches of: (a) the input CAD file; (b) the extracted polygons (colour-coded with their usage type); and (c) the wall polygons

on accessibility and safety criteria, among which accessibility is an essential requirement for ADA compliance. Two network attributes (i.e. ADA_RESTRICTED and ROOT_PREFERENCE) were added to the network dataset as 'restriction' and 'cost' types with the purpose explained earlier in 'Construction of 3D network'. After we built the network dataset, the route-finding task was performed in ArcScene using the geoprocessing model presented in Figure 10a. The model utilised a 3D network dataset, restrictions, and impedance attributes as the input parameters and found the optimal route between two user-specified 3D locations. Once the model found the optimal route, it added the resulting 3D route to the current map document in ArcScene. Figure 10b shows an optimal route layer added to ArcScene.

## 4. Example

Next, we present a series of examples from USC's University Park campus. We built 17 building models of high visual quality, one of which included 3D indoor models as well as a 3D indoor network dataset. Three exemplary routes were determined in ArcScene using the procedure presented in 'Best route finding'. The resulting routes were then modified with 'Buffer3D' in ArcToolbox, exported as multipatch shapefiles, and then imported to CityEngine as static layers. The resulting 3D web scene generated from CityEngine can be accessed by most web browsers supporting WebGL format. The CityEngine WebViewer also provides the user interface for navigating the 3D web scene interactively (Figure 11a). The 3D indoor models of the AHF building are

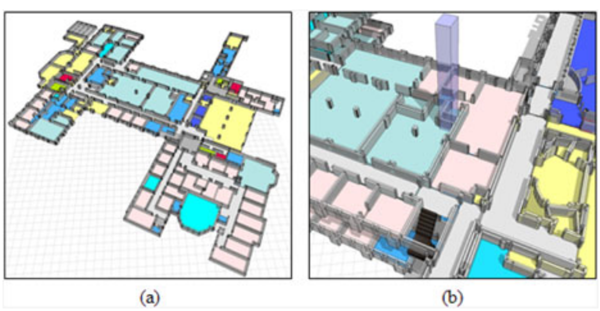


**Figure 6.** Generation of 3D indoor models: (a) 3D indoor model; and (b) CGA-generated elevator and staircase

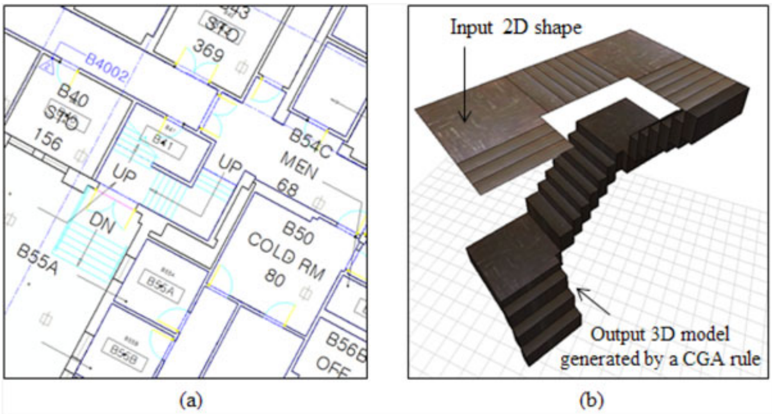


**Figure 7.** 3D staircase model: (a) Staircase in the CAD file; (b) CGA-generated 3D staircase model

shown in Figure 11b. The semantic information explained in 'Extraction of principal features' can also be queried by using the search icon or by clicking each object in the WebViewer as illustrated in Figure 11b.

Figure 12 shows three exemplary 3D routing results visualised with 3D models in the WebViewer. With regard to the route shown in Figures 12a and 12b, the destination point was chosen inside the main lobby of the Spatial Sciences Institute (SSI) located in the basement of the AHF building. Turning on the ADA_RESTRICED network attribute, the model found the route following the staircase both for entering the building and for going down to the basement. However, with the ADA_RESTRICTED turned off, the model found the wheelchair accessible route as presented in Figures 12c and 12d. Compared with the previous route shown in Figures 12a and 12b, the resulting path entered the building using the ramp instead of the staircase. It also used the elevator rather than following the staircase, even though the resulting route is not the fastest and shortest compared to the route in Figures 12a and 12b. Another ADA-compatible routing example is presented in Figures 12e and 12f, which shows a routing result inside the building. It starts from an office on the fourth floor and ends at an office in the basement. Clearly, the model found the optimal path following the elevator instead of using the staircase. The given 3D models and routing visualisation in this example can be accessed at http://cityengine.usc.edu/ceviewer.html?3dWebScene=AllanHancockFoundationBuilding_low.3ws.

## 5. Discussion and conclusion

In this article, we have presented how existing resources can be utilised to realise a 3D indoor/outdoor routing and visualisation application. As demonstrated in Section 4, the ADA-compatible 3D routes we determined could be visualised seamlessly with 3D models of both the indoor and outdoor spaces of the buildings with the help of CityEngine's visualisation capabilities. We proved that the proposed workflow was appropriate, especially for the generation of 3D indoor models and 3D route visualisation. However, we found some limitations for producing a fully operational 3D routing application at this time, as summarised below.

The current version of CityEngine, at the time of this writing, does not support the runtime generation of WebGL, which requires all the layers to be publishable to the web, including the 3D indoor and façade models and

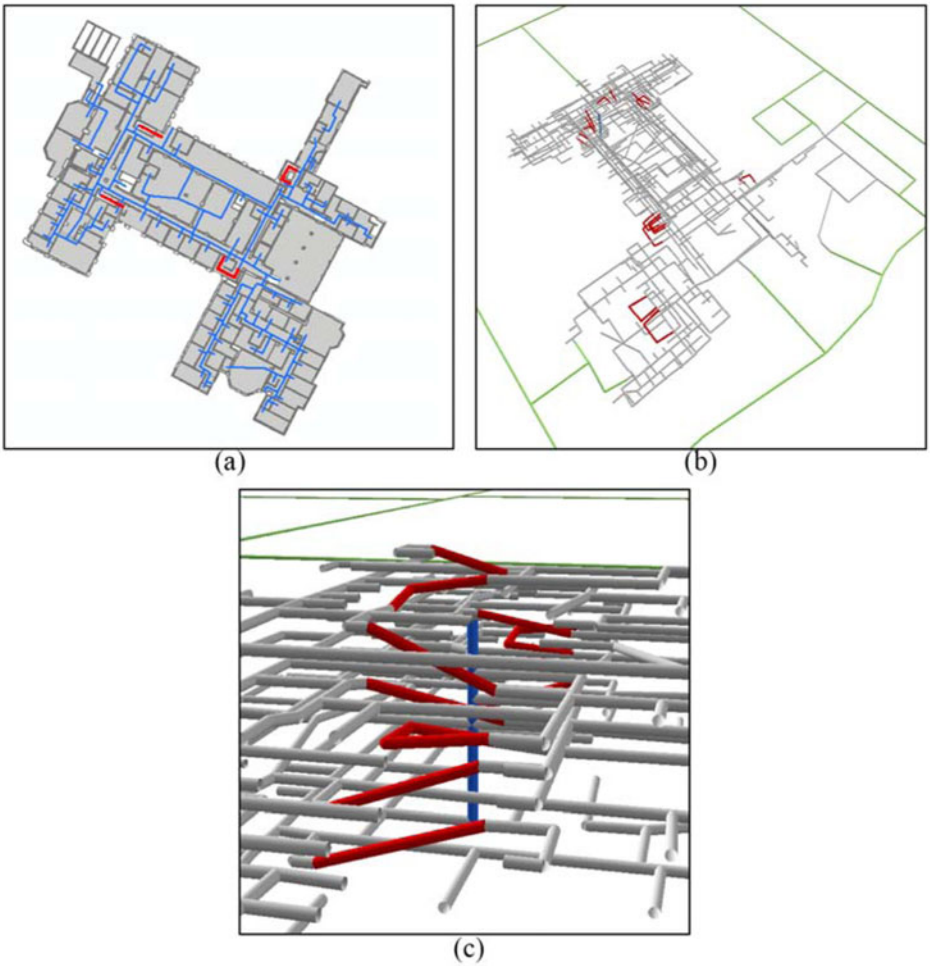


**Figure 8.** Construction of 3D network: (a) 2D floor lines (blue) and staircases (red); (b) Completed 3D multimodal network; and (c) Zoomed-in view

the routing results. All of these items should be defined within CityEngine prior to generating the final WebGL. Our testing demonstrated that the current framework was suitable for providing predefined 3D routes to the user; however, it does not allow the user to interact with the routing application. This limitation could be resolved once a new version of CityEngine is released, in which relevant functionalities as well as CityEngine's new JavaScript API are expected to be introduced.

In the context of data preparation, a considerable degree of manual processing is still required for the 3D network, indoor and façade models. As demonstrated in this article, the availability of 2D CAD floor plans can significantly reduce the time required for the generation of 3D indoor models. A Building Information Model (BIM) can be considered as another potential data source that can be exploited for this purpose. Compared to conventional GIS data, a BIM can create 3D

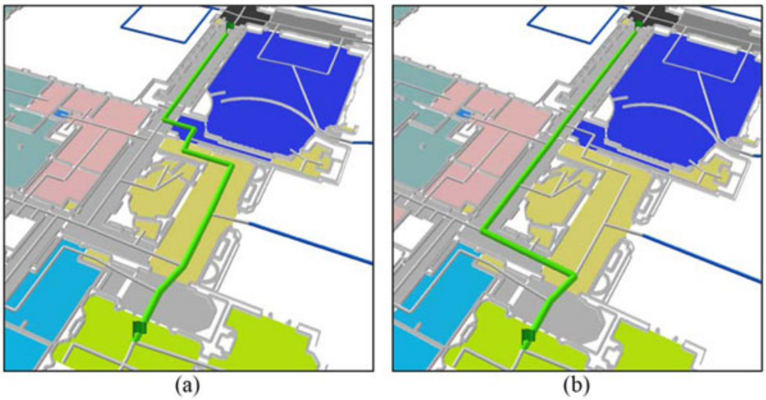


**Figure 9.** Different routing results: (a) without *PREFERENCE*; and (b) with *PREFERENCE*

Architecture, Engineering, and Construction (ACE) models, which are more focused on individual constructive elements, such as walls, spaces, windows, doors, and beams. The current version of CityEngine, however, does not support directly importing BIM data. Instead, an Industry Foundation Classes (IFC) data model can be used for sharing information

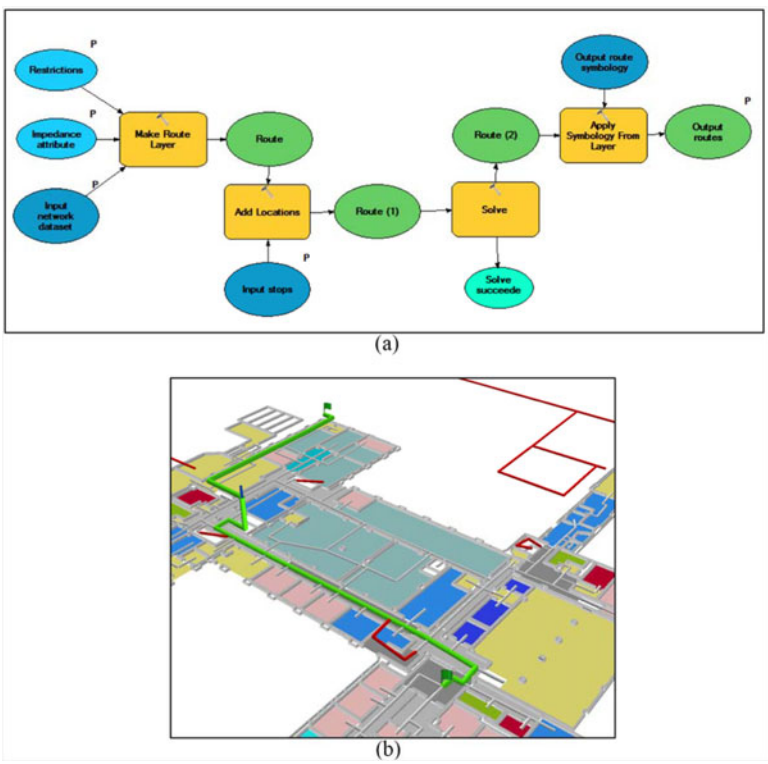


**Figure 10.** 3D route-finding task in ArcScene: (a) Geoprocessing model; and (b) Resulting route visualised with 3D indoor model

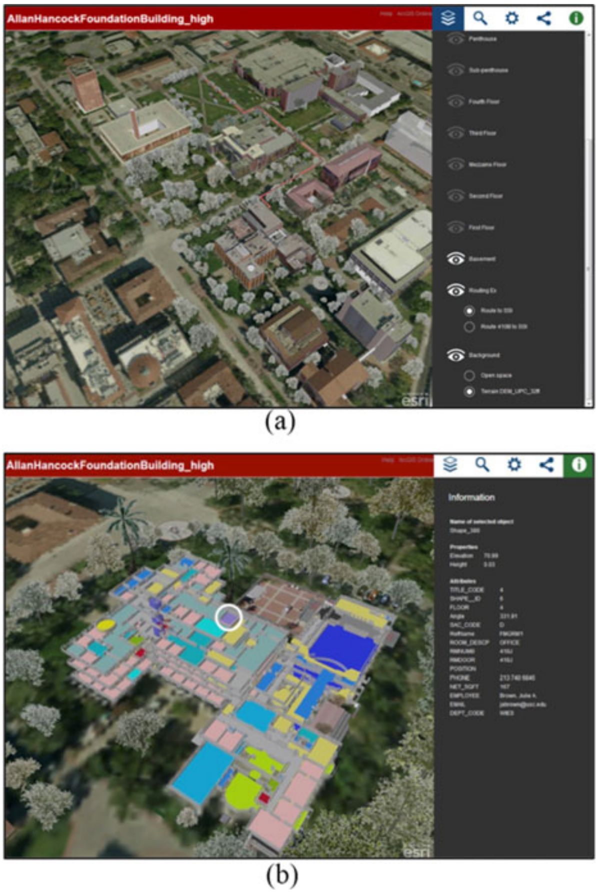


**Figure 11.** CityEngine WebViewer interface: (a) UPC 3D models; and (b) 3D indoor models of AHF building and object attributes

and spanning the gap between a BIM and GIS. For instance, IFC data exported from an original Autodesk Revit model can be transformed to an Esri file geodatabase using the Data Interoperability extension in ArcGIS, in which each layer and class represents one IFC entity and constructive element, respectively, after which the resulting geodatabase can be imported into CityEngine. Figure 13 compares the imported BIM model with our 3D model generated from CGA shape grammar. In this example, the imported BIM model consists of 16 IFC classes and represents more detailed constructive elements. Each IFC entity also inherits the object attributes defined in its original Revit data model. Despite its rich information content, both geometry and semantic information, they are not always necessary or even favourable for the specific application at hand. For example, the IFC class (e.g. IfcWallStandardCase class) shown in Figure 13c includes most of both the indoor walls and outdoor walls (i.e. building façades). Each IfcWallStandardCase entity has the following attribute format: '*Basic Wall:[Filename]-[Dimension][Material]:[Object ID]*' and the attributes of

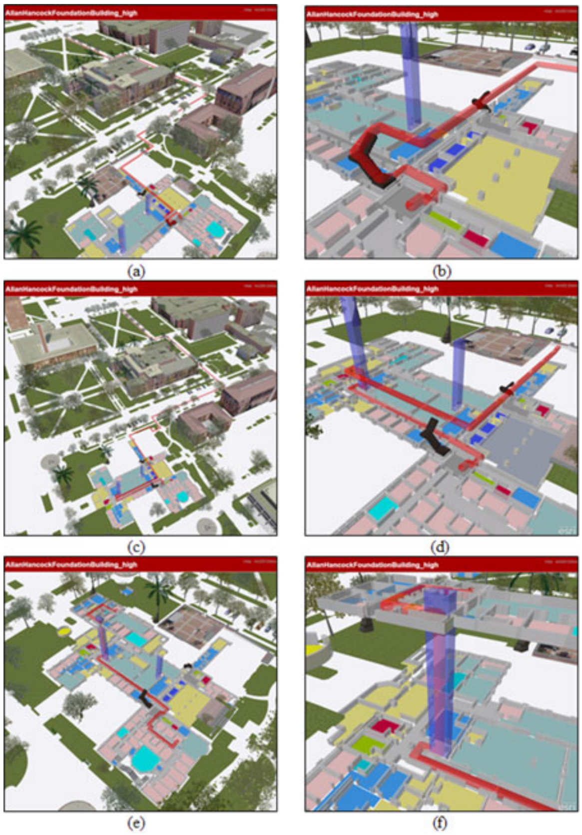

**Figure 12.** Examples of 3D routing visualisation: (a) and (b) ADA-restricted route to SSI; (c) and (d) ADA-compatible route to SSI; and (e) and (f) ADA-compatible route inside AHF building

two exemplary IFCs are presented as follows:

> Basic Wall:0409SSL-5″ Brick Wall: 164415
> Basic Wall:0409SSL-6″ Concrete Wall: 221851

Based on the given attribute, it turns out that discriminating indoor walls from building façades is not straightforward as the object attributes are more oriented to the design concepts. However, for visualisation purposes, some IFC classes can be selectively imported and used as static layers to enhance visual quality (e.g. the stairway model shown in Figure 13d).

In practice, CAD files or BIM models are not always available because either they do not exist or their accessibility is restricted. In this case, different approaches should be taken, possibly leading to more time and human effort. In this context, the use of crowd-sourced geodata could be one solution for overcoming this limitation. For instance, Open Street Map (OSM) is now expanding into indoor spaces, i.e. IndoorOSM (Goetz & Zipf 2011; Rocha & Alves 2012). As a kind of Volunteered Geographic Information (VGI), these approaches offer enormous potential, notwithstanding the fact that the data reliability and consistency are not guaranteed.

Another limitation of our study was the lack of semantic information about the safety criteria for ADA standards. In this study, we only considered the accessibility criteria to determine ADA-compatible routes. However, as declared in the ADA Standards for

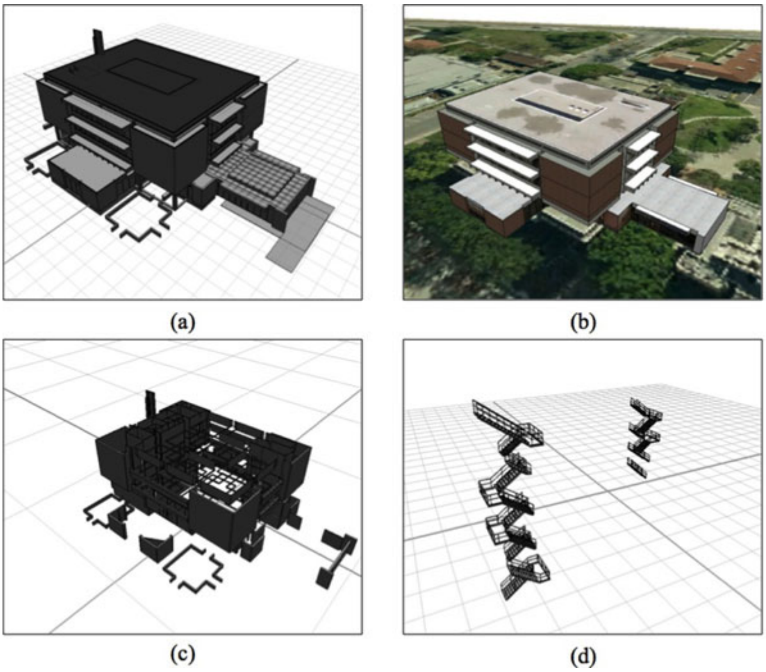


**Figure 13.** Comparison between BIM and CGA-generated model (USC Seaver Science Library): (a) Imported BIM model; (b) CGA-generated model; (c) IfcWallStandardCase class; and (d) IfcRailing and IfcStairFlight classes

Accessible Design (ADA 2010), there are a number of different requirements to support individuals with disabilities. For instance, safety-related information, such as street type, width, surface, and slope of the way or curb, clearly affects the way-finding process, resulting in different routing results with users' different preferences or physical conditions.

Despite its aforementioned limitations, we believe we demonstrated CityEngine's potential for 3D indoor routing. Compared to some of the existing 3D indoor routing services, CityEngine's output, WebGL, is a cross-platform based on the open-source library. It works on a majority of desktops as well as a growing number of mobile browsers. Therefore, a 3D web scene generated from CityEngine can be accessed anywhere by public computers and personal mobile devices. Considering the nature of the 3D indoor routing task, this application could be a big advantage. Moreover, CityEngine, as one of the software packages supported by Esri, offers full and seamless integration with numerous other geospatial platforms and tools, such as ArcGIS, ArcScene, and Network Analyst. Our work has shown how some of the existing GIS analysis functionalities can be used with CityEngine.

## Acknowledgements

The authors acknowledge and express our appreciation for the comments of the anonymous reviewers. We also thank Eric Wittner, Andreas Ulmer, and Robert Garrity from Esri for their discussions with us about CityEngine and development plans, as well as Geoff Taylor at Esri for inviting us to present our work at the 2013 Esri International User Conference in San Diego, California.

## References

ADA (2010) Americans with Disabilities Act (ADA) Standards For Accessible Design, Available from: http://www.ada.gov/2010ADAstandards_index.htm

Baillard, C., & Maıtre, H. (1999) 3-D reconstruction of urban scenes from aerial stereo imagery: a focusing strategy. *Computer Vision and Image Understanding*, vol. 76, no. 3, pp. 244–258.

Biber, P., Andreasson, H., Duckett, T., & Schilling, A. (2004) 3D modeling of indoor environments by a mobile robot with a laser scanner and panoramic camera. *Proceedings of International Conference on Intelligent Robots and Systems*, Sendai, Japan, pp. 3430–3435.

Dudas, P.M., Ghafourian, M., & Karimi, H.A. (2009) ONALIN: Ontology and algorithm for indoor routing. *Proceedings of the 10th International Conference on Mobile Data Management: Systems, Services and Middleware*, Taipei, Taiwan, pp. 720–725.

Fraser, C., Baltsavias, E., & Grün, A. (2001) 3D building reconstruction from high-resolution Ikonos stereo-imagery. *Automatic Extraction of Man-Made Objects From Aerial and Space Images (III)*, Ascona, Switzerland, pp. 331–344.

Goetz, M. (2013) Towards generating highly detailed 3D CityGML models from OpenStreetMap. *International Journal of Geographical Information Science*, vol. 27, no. 5, pp. 845–865.

Goetz, M., & Zipf, A. (2011) Extending OpenStreetMap to indoor environments: bringing volunteered geographic information to the next level. *Proceedings of the Urban and Regional Data Management*, Delft, The Netherlands, pp. 47–58.

Haala, N., & Kada, M. (2010) An update on automatic 3D building reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 65, no. 6, pp. 570–580.

Halatsch, J., Kunze, A., & Schmitt, G. (2008) Using shape grammars for master planning. *In: Gero. J.S., ed. Design Computing and Cognition'08*, Springer, New York, pp. 655–673.

Hagedorn, B., Trapp, M., Glander, T., & Dollner, J. (2009) Towards an indoor level-of-detail model for route visualization. *Proceedings of the 10th International Conference on Mobile Data Management: Systems, Services and Middleware*, Taipei, Taiwan, pp. 692–697.

Horna, S., Damiand, G., Meneveaux, D., & Bertrand, Y. (2007) Building 3D indoor scenes topology from 2D architectural plans. *Proceedings of the International Conference on Computer Graphics Theory and Applications (GRAPP)*, Barcelona, Spain, pp. 37–44.

Huang, H., Brenner, C., & Sester, M. (2011) 3D building roof reconstruction from point clouds via generative models. *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Chicago, IL, pp. 16–24.

IFCwiki.org; Website that hosts information about industry foundation classes. Available from: http://www.ifcwiki.org/index.php/Main_Page, Accessed February 25, 2014.

Karimi, H.A., & Ghafourian, M. (2010) Indoor routing for individuals with special needs and preferences. *Transactions in GIS*, vol. 14, no. 3, pp. 299–329.

Kim, K., & Shan, J. (2011) Building roof modeling from airborne laser scanning data based on level set approach. *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 66, no. 4, pp. 484–497.

Kwan, M.-P., & Lee, J. (2005) Emergency response after 9/11: the potential of real-time 3D GIS for quick emergency response in micro-spatial environments. *Computers, Environment and Urban Systems*, vol. 29, no. 2, pp. 93–113.

Li, X., Claramunt, C., & Ray, C. (2010) A grid graph-based model for the analysis of 2D indoor spaces. *Computers, Environment and Urban Systems*, vol. 34, no. 6, pp. 532–540.

Liu, J., Lyons, K., Subramanian, K., & Ribarsky, W. (2010) Semi-automated processing and routing within indoor structures for emergency response applications. *Proceedings of SPIE7709, Cyber Security, Situation Management, and Impact Assessment II; and Visual Analytics for Homeland Defense and Security II, 77090Z*; doi:10.1117/12.855001

Maas, H.G. (1999) The potential of height texture measures for the segmentation of airborne laserscanner data. *Proceedings of the 4th Ariborne Remote Sensing Conference and Exhibition*, Ottawa, Canada, pp. 21–24.

Meijers, M., Zlatanova, S., & Pfeifer, N. (2005) 3D geoinformation indoors: structuring for evacuation. *Proceedings of Next Generation 3D City Models*, Bonn, Germany, pp. 21–22.

Müller Arisona, S., Zhong, C., Huang, X., & Qin, R. (2013) Increasing detail of 3D models through combined photogrammetric and procedural modelling. *Geo-spatial Information Science*, vol. 16, no. 1, pp. 45–53.

Müller, P., Wonka, P., Haegler, S., Ulmer, A., & Van Gool, L. (2006) Procedural Modeling of Buildings, *ACM*, *Transactions on Graphics*, vol. 25, no. 3, pp. 614–623.

Parish, Y.I., & Müller, P. (2001) Procedural modeling of cities. *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, Los Angeles, CA, pp. 301–308.

Prusinkiewicz, P., & Lindenmayer, A. (1990) *The Algorithmic Beauty of Plants*, Springer-Verlag, New York.

Prusinkiewicz, P., Mündermann, L., Karwowski, R., & Lane, B. (2001) The use of positional information in the modeling of plants. *Proceedings of the 28th Annual Conference On Computer Graphics and Interactive Techniques*, Los Angeles, California, pp. 289–300.

Rocha, J., & Alves, N. (2012) OSM indoor: moving forward. Available from: http://ogrs2012.org/index.php/ogrs2012/ogrs2012/paper/view/72

Rottensteiner, F., & Briese, C. (2002) A new method for building extraction in urban areas from high-resolution LIDAR data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 34, no. 3, pp. 295–301.

Sampath, A., & Shan, J. (2010) Segmentation and reconstruction of polyhedral building roofs from aerial Lidar point clouds. *IEEE Transactions on Geoscience Remote Sensing*, vol. 48, no. 3, pp. 1554–1567.

Vosselman, G. (1999) Building reconstruction using planar faces in very high density height data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 32, no. 3, pp. 87–94.

Winter, S. (2012) Indoor spatial information. *International Journal of 3-D Information Modeling*, vol. 1, no. 1, pp. 25–42.

Wonka, P., Wimmer, M., Sillion, F., & Ribarsky, W. (2003) Instant architecture. *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 669–677.